Black Hat Europe 2017

Black Hat Europe 2017 - Author not stated, blackhat.com.

- Published: date not stated
- Original: <https://blackhat.com/archive/europe/2017/briefings.html#breaking-out-hsts>
- Preserved from: https://blackhat.com/archive/europe/2017/briefings.html#breaking-out-hsts (live) on 2026-08-08
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Black Hat Europe 2017 | Briefings

briefings

- Lillian Ablon
- Andy Applebaum
- Fernando Arnaboldi
- David Atch
- Jared Atkinson
- Arnaud Bannier
- Max Bazaliy
- Sheila Berta
- Andrés Blanco
- Pepijn Bruienne
- Sonia Burney
- Leon Böck
- Phuong Cao
- Shuo Chen
- Lei Cheng
- Sharon Conheady
- Andrea Continella
- Jake Corina
- Joshua Crumbaugh

- Stefano D'Alessio
- Sergio De Los Santos
- Adam Donenfeld
- Ayoub Elaassal
- Mark Ermolov
- Eric Filiol
- Lorenzo Fontana
- Maxim Goryachy
- Fabio Gritti
- Avital Grushcovski
- Lion Gu
- Shuang Hao
- Saurabh Harit
- Sean Heelan
- Ravishankar Iyer
- Min-Chang Jang
- Shankar Karuppayah
- Eugene Kogan
- John Kozyrakis
- Vladimir Kropotov
- Kyoung-ju Kwak
- George Lashenko
- Antti Levomäki
- Tal Liberman
- Andreas Lindh
- HC MA
- Aravind Machiry
- Debasish Mandal
- Sebastiano Mariani
- Brent Maynard
- Matt McCutchen
- Daniel Mende
- Maxime Meyer
- Douglas Miller
- Jeff Moss
- Shawn Moyer
- Collin Mulliner
- Max Mühlhäuser
- Olli-Pekka Niemi

- Chris Painter
- Mario Polino
- Shaz Qadeer
- Loïc Rouch
- Joanna Rutkowska
- Chris Salls
- Hendrik Schmidt
- Ory Segal
- Ben Seri
- Simha Sethumadhavan
- Chi-en (Ashley) Shen
- Yan Shoshitaishvili
- Elad Shuster
- Dmitry Sklyarov
- Rich Smith
- Salvatore Stolfo
- Adrian Tang
- Chris Thompson
- Mathy Vanhoef
- Emmanouil Vasilomanolakis
- Gregory Vishnepolsky
- Robby Winchester
- Jan Helge Wolf
- David Wong
- Fyodor Yarochkin
- Stefano Zanero
- Yakun Zhang

briefings - December 6 & 7

Keynote

Diplomacy and Combating Evolving International Cyber Threats

Governments and high-level executives have transitioned from seeing both policy and technical cyber threats as solely technical issues to core issues of national security, economic policy, human rights and, ultimately, foreign policy. Drawing on experience at the U.S. Department of Justice, at the White House, and finally at the U.S. State Department, I will detail the evolving cyber landscape and growing efforts to counter cyber threats internationally. I will focus on the emerging field of

international "cyber diplomacy" and reveal how these tools have been used in particular cases as well as to counter larger policy threats posed by repressive regimes. I'll also outline efforts to promote norms (rules of the road) in cyberspace and the role of attribution and deterrence. Finally, I will examine challenges that lie ahead and the need for the policy and technical communities to work together globally to meet those challenges.

Presented By

Chris Painter

Security Through Distrusting

There are different approaches to making (computer) systems (reasonably) secure and trustworthy:

At one extreme, we would like to ensure everything (software, hardware, infrastructure) is *trusted*. This means the code has no bugs or backdoors, patches are always available and deployed, admins always competent and trustworthy, and the infrastructure always reliable...

On the other end of the spectrum, however, we would like to *distrust* (nearly) all components and actors, and have no single almighty element in the system.

In my opinion, the industry has been way too much focused on this first approach, which I see as overly naive and non-scalable to more complex systems.

In this talk, based on my prior work as both offensive researcher in the past, as well as an engineer and architect on the defense side in the recent years, I will attempt to convince the audience that moving somehow towards the "security through distrusting" principle might be a good idea. Equally important though, the talk will discuss the trade-offs that this move requires and where can we find the sweet spot between the two approaches.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Rutkowska-Security-Through-Distrusting.pdf)

Presented By

Joanna Rutkowska

Briefings

A Process is No One: Hunting for Token Manipulation

Does your organization want to start Threat Hunting, but you're not sure how to begin? Most people start with collecting ALL THE DATA, but data means nothing if you're not able to analyze it properly. This talk begins with the often overlooked first step of hunt hypothesis generation which

can help guide targeted collection and analysis of forensic artifacts. We will demonstrate how to use the MITRE ATTACK Framework and our five-phase Hypothesis Generation Process to develop actionable hunt processes, narrowing the scope of your Hunt operation and avoiding "analysis paralysis." We will then walk through a detailed case study of detecting access token impersonation/manipulation from concept to technical execution by way of the Hypothesis Generation Process. Along the way, we will detail some of the most common access token manipulations in use and detail the defensive detection implications for each of these cases. This comprehensive case study will better arm both attackers and defenders with how to better utilize their toolset to detect or avoid detection of token theft and manipulation.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Atkinson-A-Process-Is-No-One-Hunting-For-Token-Manipulation.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Atkinson-A-Process-Is-No-One-Hunting-For-Token-Manipulation-wp.pdf)

Presented By

Jared Atkinson & Robby Winchester

A Universal Controller to Take Over a Z-Wave Network

With the advent of Internet-of-Things, Z-Wave is a major communication protocol for home automation systems. Z-Wave devices have to satisfy end-users convenience and thus are plug-and-play. Additionally, high market competitiveness leads to short development cycle, pushing aside security requirements.

Indeed, Z-Wave+ standard (5th generation), including stronger security with encryption, has been lately adopted. However, many deployed Z-Wave devices do not support this new version. Furthermore, Z-Wave+ devices have to be backward compatible and thus support both secure and insecure modes. Since users are primarily looking for functionality and using devices as plug-n-play, documentation is overseen and co-existence of two modes may lead to misunderstandings while secure mode is usually deactivated by default to ease the installation.

What we called an insecure mode is a security relying on the uniqueness of the controller HomeID (network identifier) which is supposedly not alterable (in official equipment). However, with dedicated equipment (Software-Defined Radio), attackers can alleviate such a limitation (Black Hat 2014, Picod et. al). Purchasing and the difficulty to use this type of equipment limit the threat to expert attackers. Imagine now that a simple device can be used for the same purposes, by any ill-intentioned person (from the unpleasant neighbors to the common thief opening access to the home).

In this talk, we will show that using only an official and cheap mainstream device, taking over a full network is possible. We rely on a standard feature of Z-Wave (auto-discovery) and on additional functionality of an official controller (backup/restore). Both are legitimate but combined together they allow to create a universal controller by pre-filling all device identifiers in advance (without passive listening). As a result, all devices can be controlled. If a user add a new one, it will be automatically controlled by our controller as well.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Rouch-A-Universal-Controller-To-Take-Over-A-Z-Wave-Network.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Rouch-A-Universal-Controller-To-Take-Over-A-Z-Wave-Network-wp.pdf)

Presented By

Loïc Rouch

Attacking NextGen Roaming Networks

Weaknesses of SS7 Roaming Networks are well known – but what about the Diameter interfaces coming up at the moment? Diameter is and will be used for roaming connections of LTE/LTE-A mobile networks - a new architecture, and a new implementation. But still, one remains the same: it is a AAA protocol designed for trusted environments - roaming interconnection interfaces between providers.

As we know from the past, it is possible to get access to such networks, as you can simply buy access if you spend enough money; as typical attackers in such environments are fraudsters or agencies, they definitely will. Therefore, securing these interface and assessing the infrastructure components and its configuration is very important.

In our talk, we will explain not only how Diameter-based networks work and which messages and functions exit, but also which of them can be abused by attackers. Typical attacks are information leaks about the environment, but also attacks against the authentication and encryption of customers. These information can be used for interception of mobile data/calls, but also to establish new business models of fraud.

To demonstrate such attacks, we developed a testing framework covering information gathering, mobile phone tracking, denial of service attacks, pay fraud, and interception of data. The framework will be released during our talk and will enable providers and security companies to assess a telco's diameter network configuration and demonstrate what can happen if no proper security measures are applied. We also will give an outlook on how a provider can protect from such kind of attacks.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Schmidt-Attacking-Next-Gen-Roaming-Networks.pdf)

Presented By

Hendrik Schmidt & Daniel Mende

Attacks Against GSMA's M2M Remote Provisioning

GSMA is investigating, developing and standardizing an embedded SIM card with remote provisioning (that is, over-the-air installation of subscription data), called an eUICC, to improve the current mobile-phone subscription model. In this presentation, we will review remote-provisioning

security mechanisms and show that these mechanisms are vulnerable to attacks that prevent network operators from providing service, in particular, we:

• Identify three classes of attacks by malicious insiders that prevent operators from installing subscription data on eUICC's; and

• A further attack by a network adversary that exhausts an eUICC's memory

These attacks arise from flaws in the specification, and we will discuss fixes that will improve security for next generation telecommunication networks. The presentation will include insights to the specification that are not yet public. It will also include GSMA's reaction to our findings. The presentation is based on research by Maxime Meyer, Elizabeth Quaglia and Ben Smyth, and it is supported by a detailed technical report, which will be released after the presentation.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Meyer-Attacks-Against-GSMAS-M2M-Remote-Provisioning.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Meyer-Attacks-Against-GSMAS-M2M-Remote-Provisioning-wp.pdf)

Presented By

Maxime Meyer

Automatic Discovery of Evasion Vulnerabilities Using Targeted Protocol Fuzzing

Network protocol normalization and reassembly is the basis of traffic inspection performed by NGFW and IPS devices. Even common network protocols are complex with multiple possible interpretations for the same traffic sequence. We present a novel method for automated discovery of errors in traffic normalization by targeted protocol stack fuzzing. These errors can be used by attackers to evade detection and bypass security devices. We will demonstrate the techniques against up-to-date security devices and show that many security devices still have basic evasion vulnerabilities. The tools used will be publicly available after this presentation.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Levomaki-Automatic-Discovery-Of-Evasion-Vulnerabilities-Using-Targeted-Protocol-Fuzzing.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Levomaki-Automatic-Discovery-Of-Evasion-Vulnerabilities-Using-Targeted-Protocol-Fuzzing-wp.pdf)

Presented By

Antti Levomäki & Olli-Pekka Niemi

Becoming You: A Glimpse into Credential Abuse

In recent years, security threats have exponentially increased, as have the potential solutions to detect and mitigate these threats. The question often is, where in the workflow to deploy

respective detection and mitigation strategies that are risk averse while maintaining efficacy. For a given strategy, the idea is to observe the request workflow to determine how the request presents itself in terms of headers and other attributes that describe the request, how the client processes the challenge questions, and how the request is managed accordingly. Detection is managing things we know such as partner bots, aggregators, and search engines. Detection strategies include both request and session anomalies, and more specifically, session tracking, rate detection, workflow validation, behavioral detection, fingerprint anomalies, header anomalies, and cookie validation. Mitigation strategies include alternate actions such as static content and/or honey pots, deny, request rate modification, and tarpit actions using session black-holing. In this talk, we discuss how to detect and mitigate vulnerabilities using CDNs or existing back end platform architecture. CDN based technology helps to offload security protection during request flow versus relying solely on the origin. With or without CDN specific functionality, platform techniques helps to manage security at the earliest point in the request flow across all nodes within the back end architecture in order to correlate data amongst nodes based on attributes and reputational intelligence to provide actionable data to the system.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Burney-Becoming-You-A-Glimpse-Into-Credential-Abuse.pdf)

Presented By

Sonia Burney & Brent Maynard

BlueBorne - A New Class of Airborne Attacks that can Remotely Compromise Any Linux/IoT Device

The thought of a cyber attack spreading through the air like a plague was only a distant fear, until recently. This year, the airborne attack surface emerged, with significant vulnerabilities found in prominent wireless protocols - first Wi-Fi, and now Bluetooth, in the recently identified BlueBorne attack. In this talk we will present the ramifications of airborne attacks, which bypass all current security measures and provide hackers with a contagious attack, capable of jumping over "air-gapped" networks and allow easy full remote code execution on devices from every major OS. We will demonstrate two out of eight zero-day exploits we've found in the Bluetooth stacks of Linux, Android, Windows, and iOS. We will show how easy it is to infect devices ranging from ordinary PCs to the emerging realm of IoT devices, requiring no user interaction, and nothing more than an active Bluetooth.

We will present our findings in Linux, and display a step-by-step exploitation process providing full control over any device running Linux, or any OS derived from it, which unfortunately, includes the majority of IoT devices. We will also explain how to create a generic exploit that can be adapted to operate on different devices and architectures.

Our talk will emphasize that real threats in cyber security are hiding in plain sight. BlueBorne exposes vulnerabilities in Bluetooth implementations, a protocol which has been around since 1998. While researchers did tackle the protocol's flaws, the potential of vulnerabilities in widespread Bluetooth stacks which affect devices directly has been overlooked for the past

decade. Bluetooth might seem peripheral, but it holds a large prize for attackers, since it runs on the core of a device with high privileges. This demonstrates the devastating effect of Bluetooth's combined traits: a less scrutinized protocol, that once exploited can provide full control over targeted devices.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Seri-BlueBorne-A-New-Class-Of-Airborne-Attacks-Compromising-Any-Bluetooth-Enabled-Linux-IoT-Device.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Seri-BlueBorne-A-New-Class-Of-Airborne-Attacks-Compromising-Any-Bluetooth-Enabled-Linux-IoT-Device-wp.pdf)

Presented By

Ben Seri & Gregory Vishnepolsky

Breaking Bad: Stealing Patient Data Through Medical Devices

This talk discusses the risks of connected healthcare devices. It looks at the benefits of adopting IoT for medical devices, current exposure, common communication channels in use as well as interconnectivity approaches used with other critical components. Based off output from security assessments performed against medical devices widely deployed at various hospitals and medical institutions, I will present an in-depth analysis of the target medical device and elaborate on how I was able to compromise it to gain access to plethora of medical records from all the medical institutions it was deployed at and not just the one where our target device was hosted.

I will introduce the threat surface exposed by various medical devices and present some of the real-world attacks against some popular devices & their impact on humans as well as the overall ecosystem they are connected to. Some devices rely on proprietary hardware on licensed bands, which reduces the risk of interference from consumer connected devices, but doesn't provide security as implied in marketing materials. Others rely on standard Wi-Fi security measures for confidentiality and are prone to MitM attacks. Healthcare devices that implement IrDA could yield interesting results when interfaced with cheap $10 hardware.

There are many consumer items that fall under the umbrella of IoT and while it may be hard to understand the impact of hacking a toaster, we can all agree that manipulation of a medical device could lead to rather serious consequences. Apart from putting a patient's life at risk, an attacker could compromise a healthcare device to steal patient data. This presentation will primarily focus on the latter with real-world examples and a case study. I will demonstrate the compromise of a healthcare device to steal medical records, which typically include PII, health insurance data, medical history, SSNs, prescriptions etc.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Harit-Breaking-Bad-Stealing-Patient-Data-Through-Medical-Devices.pdf)

Presented By

Saurabh Harit

Breaking Out HSTS (and HPKP) on Firefox IE/Edge and (Possibly) Chrome

Observing Microsoft's reports, the Edge browser - and its previous version, Internet Explorer - supports HSTS (HTTP Strict Transport Security) beginning with IE 11 over Windows 7, 8.1 and 10. However, official technical documentation does not exist about how this system works in the browser, how the data is saved nor any other information. Likewise, there is no official documentation about how Firefox and Chrome implement it.

Despite this, the truth is that for an attacker, techniques like SSLStrip stopped being fully effective after HSTS and HPKP implementation. A remote attack against HSTS named Delorean was presented some time ago, but it has some limitations. Throughout our research, we have discovered new attacks and new inconsistencies in the web browsers when solving issues related to HSTS and HPKP.

Firefox browser has an implementation issue, for which we developed an attack that allows remotely overwriting the storage of the websites with HSTS/HPKP directives. In this way, an attacker can easily take advantage of it during a Sniffing/MITM over the LAN Network and obtain plain text credentials from sites that had set up their communication strictly over HTTPS. Chrome in the same way suffers implementation issues that can hinder notably the use of HSTS/HPKP in the browser.

We looked into IE/Edge, the runtime implementation of the API HttpIsHostHstsEnabled from WININET.DLL, gaining the knowledge to know how the invoked methods that resolve domains with HSTS works. Additionally, we found interesting considerations into the storage system (ESE Database) and several implementation issues.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Berta-Breaking-Out-HSTS-And-HPKP-On-Firefox-IE-Edge-And-Possibly-Chrome.pdf)

Presented By

Sheila Berta & Sergio De Los Santos

By-design Backdooring of Encryption System - Can We Trust Foreign Encryption Algorithms

Recent years have shown that more than ever governments and intelligence agencies strive to control and bypass the cryptographic means used for the protection of data and communications. Backdooring encryption algorithms is considered as the best way to enforce cryptographic control. Until now, only implementation backdoors (at the protocol/implementation/management level) are generally considered. In this paper we propose to address the most critical issue of backdoors: mathematical backdoors or by-design backdoors, which are put directly at the mathematical design of the encryption algorithm. While the algorithm may be totally public, proving that there is a backdoor, identifying it and exploiting it, may be an intractable problem.

We intend to explain that it is possible to design and put such backdoors. Considering a particular family (among all the possible ones), we present BEA-1, a block cipher algorithm which is similar to the AES and which contains a mathematical backdoor enabling an operational and effective cryptanalysis. Without the knowledge of our backdoor, BEA-1 has successfully passed all the statistical tests and cryptographic analyses that NIST and NSA officially consider for cryptographic validation. In particular, the BEA-1 algorithm (80-bit block size, 120-bit key, 11 rounds) is designed to resist to linear and differential ctyptanalyses. Our algorithm has been made public in February 2017 and no one has proved that the backdoor is easily detectable and have shown how to exploit it.

In the second part of this talk, we reveal which backdoor has been built, how to exploit it thus allowing to recover the 120-bit key in around 10 secondes with only 600 kb of data (300 Kb of plaintexts + 300 Kb of corresponding ciphertexts). In the final part, we addressed other ideas which are worth considering to built more complex backdoors and we will outline the possible trends in this domain.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Filiol-By-Design-Backdooring-Of-Encryption-System-Can-We-Trust-Foreign-Encryption-Algorithms.pdf)

Presented By

Eric Filiol & Arnaud Bannier

CALDERA: Automating Adversary Emulation

Adversarial assessment of a network is a critical part of securing and hardening it; done successfully, an adversarial assessment will replicate the techniques of an adversary in a realistic way. Instead of exclusively leveraging exploits, real adversaries tend to take advantage of existing, benign system functionality during their post-compromise operations. This behavior is codified in MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK;); a knowledge base of post-compromise actions of advanced persistent threats. ATT&CK; shifts the defensive focus from software patch levels, security controls, and known threat indicators to understanding and defending against common adversary behaviors.

CALDERA is a tool that can perform automated adversarial assessments against Windows enterprise networks, requiring zero prior knowledge about the environment to run. CALDERA works by leveraging its built in semantic model for how Windows enterprise domains are structured, an adversary model describing an attacker's goals and actions, and an artificially intelligent planner that makes decisions about which actions to perform. CALDERA does this all with real side effects: CALDERA features a RAT that performs adversary actions on infected hosts and copies itself over the network to increase its foothold. To most realistically emulate an adversary, CALDERA's model uses common Windows domain elements -- users, shares, credentials -- and features a library of executable techniques curated from ATT&CK;, including favorites such as running Mimikatz to dump credentials and remote execution with WMI.

As a fully automated tool, defenders can use CALDERA to verify their defenses are working appropriately and as a resource to test defensive tools and analytics. Additionally, CALDERA's

modular design allows users to customize each individual operation and provides a flexible logic so that users can incorporate their own techniques into CALDERA's automated assessments.

This talk describes CALDERA in depth, covering use cases for defenders and a demo.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Miller-CALDERA-Automating-Adversary-Emulation.pdf)

Presented By

Douglas Miller & Andy Applebaum

CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management

The need for power and energy-efficient computing has resulted in aggressive cooperative hardware-software energy management mechanisms on modern commodity devices. Most systems today, for example, allow software to control the frequency and voltage of the underlying hardware at a very fine granularity to extend battery life. Despite their benefits, these software-exposed energy management mechanisms pose grave security implications that have not been studied before.

In this talk, we present the CLKSCREW attack, a new class of software-based fault attacks that exploit the security-obliviousness of energy management mechanisms to break security. A novel benefit for the attackers is that these fault attacks become more accessible since they can now be conducted without the need for physical access to the devices or fault injection equipment. We demonstrate CLKSCREW on commodity ARM/Android devices. We show that a malicious kernel driver (1) can extract secret cryptographic keys from Trustzone, and (2) can escalate its privileges by loading self-signed code into Trustzone. As the first work to show the security ramifications of energy management mechanisms, we urge the community to re-examine these security-oblivious designs.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Tang-Clkscrew-Exposing-The-Perils-Of-Security-Oblivious-Energy-Management.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Tang-Clkscrew-Exposing-The-Perils-Of-Security-Oblivious-Energy-Management-wp.pdf)

Presented By

Adrian Tang & Simha Sethumadhavan & Salvatore Stolfo

Dealing the Perfect Hand - Shuffling Memory Blocks on z/OS

Follow me on a journey where we pwn one of the most secure platforms on earth. A giant mammoth that still powers the most critical business functions around the world: The Mainframe! Be it a wire transfer, an ATM withdrawal, or a flight booking, you can be sure that you've used the

trusted services of a Mainframe at least once during the last 24 hours. In this talk, I will present methods of privilege escalation on IBM z/OS: How to leverage a simple access to achieve total control over the machine and impersonate other users. If you are interested in mainframes or merely curious to see a what memory manipulation looks like on z/OS, you are welcome to tag along.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Elaassal-Dealing-The-Perfect-Hand-Shuffling-Memory-Blocks-On-ZOS.pdf)

Presented By

Ayoub Elaassal

DIFUZZing Android Kernel Drivers

As the rest of the Android security infrastructure improves, the Android/Linux kernel is well on its way to becoming the "weakest link," being responsible for a higher and higher fraction of vulnerabilities [1]. Most of these vulnerabilities are in kernel driver code, as this driver code is often maintained by third parties and gets less scrutiny than the kernel itself.

Specifically, most of these bugs are in driver ioctl functions [2]. Despite significant advances in automatic analysis of kernel code, current state-of-the-art tools like Syzkaller [3] and trinity [4] fail to find these bugs. This is because ioctls do not have a standard interface, and each ioctl for each driver expects different commands and data structures. The amount of manual effort required to bridge this "interface gap" for Syzkaller and trinity has hampered effort to find, pwn, and fix these issues.

The problem needed to be fixed, and so we created DIFUZE, a lightweight (yet powerful), python based interface-aware fuzzing framework for driver ioctls. DIFUZE uses a novel combination of static analysis techniques (using LLVM) to extract the structure of argument data of the ioctls from the GPL-mandated headers of kernel drivers, and uses this information to effectively fuzz drivers on the target device.

We will publish the scientific details behind DIFUZE at the ACM Conference on Computer and Communication Security (CCS), one of the premier venues in the scientific security community.

DIFUZE works. We found 32 zero-days in seven modern android phones including the Google Pixel XL. We are certain that more bugs are lurking in more phones, so we are open-sourcing the end-to-end automated tool for the public good. DIFUZE is completely automated -- just give it kernel.tar.gz, wait, and collect the 0days.

Happy hunting.

[1] https://source.android.com/security/bulletin/; [2] Jeffrey Vander Stoep. 2016. Android: protecting the kernel. In Linux Security Summit. Linux Foundation; [3] Google. 2017. syzkaller - linux syscall fuzzer. (2017). https://github . com/google/syzkaller; [4] Dave Jones. 2011. Trinity: A system call fuzzer. In Proceedings of the 13th Ottawa Linux Symposium, pages.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Corina-Difuzzing-Android-Kernel-Drivers.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Corina-Difuzzing-Android-Kernel-Drivers-wp.pdf)

Presented By

Aravind Machiry & Chris Salls & Yan Shoshitaishvili & Shuang Hao & Jake Corina

---

Enraptured Minds: Strategic Gaming of Cognitive Mindhacks

There is no spoon! We live in the Matrix and no information can be trusted without validation. Content promotion services have been available in the grey market for some time, but the Fake News Phenomena did not gain public attention until a number of media outlets argued that several elections were heavily influenced by fake news. In this presentation, we will demonstrate tools and methods available to public mind manipulators, and walk through a selection of previous research, including USSR's study on public control and a Chinese study on crisis management. Moreover, we take Twitter posts as examples to demonstrate how significant social events are used by malefactors to promote their own agenda. We will also demonstrate a number of techniques used by cyber propaganda perpetrators, as well as how to detect their activities. To conclude, we will visit several underground markets to see how the services facilitate opinion manipulation.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Kropotov-Enraptured-Minds-Strategic-Gaming-Of-Cognitive-Mindhacks.pdf)

Presented By

Vladimir Kropotov & Fyodor Yarochkin & Lion Gu

Exfiltrating Reconnaissance Data from Air-Gapped ICS/SCADA Networks

Air-gapped industrial networks are assumed to be impenetrable because they are disconnected from the Internet and even from corporate IT networks. However, there are multiple ways that attackers can deploy malware to an air-gapped network, including compromising vendor update mechanisms or infecting USB drives or laptops of third-party contractors who connect directly to the air-gapped network for maintenance purposes.

In this talk, we cover the following scenario: An attacker compromises the air-gapped network with autonomous, self-directed malware that performs reconnaissance to discover the network topology, the specific types of industrial devices connected to it (as with the CrashOverride malware used in the 2016 Ukrainian grid attack), and perhaps sensitive IP such as secret formulas and nuclear blueprints. Once the reconnaissance information has been collected, how do you exfiltrate the data so it can used to plan and mount physical attacks?

Previous researchers have shown how to exfiltrate data from air-gapped networks using RF signals emitted from PCs, but persistent PC-based malware has a high probability of being detected. However, Programmable Logic Controllers (PLCs) don't use anti-malware programs because they have limited CPU/memory and run embedded real-time operating systems. As a result, they're ideal targets for compromise using malicious ladder logic (the code used in PLCs).

We'll explain how to inject specially-crafted ladder logic code into a Siemens S7-1200 PLC. The code uses memory copy operations to generate frequency-modulated RF signals slightly below the AM band (340kHz-420kHz), with the modulation representing encoded reconnaissance data. The signal can then be picked up by a nearby antenna and decoded using a low-cost Software-Defined Radio (SDR) and a PC. The receiving equipment can be located just outside the facility or even mounted on a drone flying overhead.

Finally, we'll show a live demo and discuss various ways to defend against this type of attack.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Atch-Exfiltrating-Reconnaissance-Data-From-Air-Gapped-Ics-Scada-Networks.pdf)

Presented By

David Atch & George Lashenko

Exposing Hidden Exploitable Behaviors in Programming Languages Using Differential Fuzzing

Securely developed applications may have unidentified vulnerabilities in the underlying programming languages. Attackers can target these programming language flaws to alter applications' behavior. This means applications are only as secure as the programming languages parsing the code.

A differential fuzzing framework was created to detect dangerous and unusual behaviors in similar software implementations. Multiple implementations of the top five interpreted programming languages were tested: JavaScript, Perl, PHP, Python, and Ruby. After fuzzing the default libraries and built-in functions, several dangerous behaviors were automatically identified.

This paper reveals the most serious vulnerabilities found in each language. It includes practical examples identifying which undocumented functions could allow OS command execution, when sensitive file contents may be partially exposed in error messages, how native code is being unexpectedly interpreted – locally and remotely – and when constant's names could be used as regular strings for OS command execution.

The vulnerabilities, methodology, and fuzzer will be made open source, and the accompanying talk will include live demonstrations.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Arnaboldi-Exposing-Hidden-Exploitable-Behaviors-In-Programming-Languages-Using-Differential-Fuzzing.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Arnaboldi-Exposing-Hidden-Exploitable-Behaviors-In-Programming-Languages-Using-Differential-Fuzzing-wp.pdf)

Presented By

Fernando Arnaboldi

Fed Up Getting Shattered and Log Jammed? A New Generation of Crypto is Coming

The SHA-3 standard came out in 2015 including the new hash function SHA-3 itself (based on a sponge construction) as well as SHAKE, a new kind of hash function called XOF. This talk will first go over these two modern algorithms, and will then introduce the other standardized functions derived from SHA-3 (KMAC, TupleHash, ParallelHash) and the more mature, modern and faster hash functions based on SHA-3: KangarooTwelve and MarsupilamiFourteen. The second part of the talk will focus on cryptographic protocols created out of SHA-3: Strobe, a symmetric protocol to protect traffic which only relies on SHA-3 as its core cryptographic function; Disco, a Noise (TLS-like) protocol and library leveraging SHA-3's properties to minimize the number of cryptographic primitives, reduce the code size, simplify the logic and increase the capabilities (hashing, generation of random numbers, derivation of keys, signing, encryption, authentication).

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Wong-Fed-Up-Getting-Shattered-And-Log-Jammed-A-New-Generation-Of-Crypto-Is-Coming.pdf)

Presented By

David Wong

The European Union's General Data Protection Regulation (GDPR) is set to go into effect in a matter of months, and already it is having a profound effect. Under GDPR rules, companies that collect or store data belonging to EU citizens or entities are required to provide top-notch privacy and security to protect that data; otherwise they could could face huge fines – as large as €20m.

As a result, companies that collect or store data are working to meet GDPR compliance. But some things are out of their control – among them third-party scripts that almost all websites depend upon to provide essential web services. Those scripts are controlled by third-parties, who may not be GDPR-compliant themselves.

Under GDPR rules, they may get fined – but the site that hosted the script is responsible too, and itself could face fines if a hacker compromises those scripts, hijacking data, installing keyloggers, etc. It's far from an uncommon problem; Banks, e-commerce sites, publishers, HMOs, insurance firms, and many others have unwittingly taken on partners whose scripts provide social media, e-commerce, advertising, content, analytics, and more – thus 'owning' their partners' security risks, too.

There have been many attempts to identify these breaches, from isolating scripts inside iFrames to scanning websites remotely using robots, to code review prior to implementation, but none of these have eliminated the problem. We propose a system where the script's actions could be isolated, and executed in an isolated environment before it is allowed to act on a "live" page. A

security system would examine the script's actions; if it acts as expected, it is allowed to apply its execution to the actual page, and if not, it remains isolated and the page remains unaffected by its payload. Thus can administrators protect themselves and avoid violating GDPR rules.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Grushcovski-GDPR-And-Third-Party-JS-Can-It-Be-Done.pdf)

Presented By

Avital Grushcovski

Heap Layout Optimisation for Exploitation

Heap-based overflows and underflows are a common vulnerability in software built in C and C++. When leveraging such a vulnerability as part of an exploit, an attacker will usually try to position the chunk which is the source of the overflow or underflow relative to a specific victim chunk. The less collateral data between the source chunk and the victim chunk, the more optimal the layout. This is typically a problem that is solved manually and requires an in-depth understanding of the heap allocator in use and how the target application interacts with it.

In this talk, we will present an automatic, blackbox, approach to heap layout optimisation. Our algorithm utilises pseudo-random search over the interactions with the allocator which may be triggered via a target application. Crucially, no modification or analysis of the allocator itself is required. Our evaluation demonstrates that this is a feasible approach on both free-list based allocators (e.g. dlmalloc) and slab based allocators (e.g. jemalloc), using heap starting states and allocator interaction sequences from a number of real world applications, including PHP, Python and Ruby.

We will also present a proof-of-concept implementation versus PHP which demonstrates that an existing fuzzer can be repurposed to perform this search. The proof-of-concept takes as input a trigger for a known vulnerability. It figures out how to interact with the allocator via PHP's API, as well as how to allocate 'useful' targets for corruption, e.g. data structures containing pointers. It produces a new PHP script as output which triggers the original vulnerability but now corrupts a specific victim chunk which it has selected. The PoC demonstrates that our approach can significantly reduce the amount of time and manual effort required to go from a crashing input to a read, write or execute primitive.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Heelan-Heap-Layout-Optimisation-For-Exploitation.pdf)[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Heelan-Heap-Layout-Optimisation-For-Exploitation-wp.pdf)

Presented By

Sean Heelan

Hiding Pin's Artifacts to Defeat Evasive Malware

Malware authors constantly develop new techniques in order to evade analysis systems. Previous works addressed attempts to evade analysis by means of anti-sandboxing and anti-virtualization techniques, for example proposing to run samples on bare-metal. However, state-of-the-art bare-metal tools fail to provide richness and completeness in the results of the analysis. In this context, Dynamic Binary Instrumentation (DBI) tools have become popular in the analysis of new malware samples because of the deep control they guarantee over the instrumented binary. In fact, in some specific scenarios (e.g., manual and automated reverse engineering) we need to fully monitor and control the analyzed binary.

As a consequence, malware authors developed new techniques, called anti-instrumentation, aimed at detecting if a sample is being instrumented. Such techniques look at the artifacts produced during the instrumentation process and leverage some intrinsic characteristics of a DBI tool.

We propose a practical approach to make DBI tools stealthier and resilient against anti-instrumentation attacks. We studied the common techniques used by malware to detect the presence of a DBI tool, and we proposed a set of countermeasures to defeat them. We implemented our approach in Arancino, on top of the Intel Pin framework. Arancino is able to hide Pin's artifacts making hard for malware to spot its presence. In order to achieve this, we leverage the power of DBI tools to fully control the execution flow of the instrumented process. This allows us to detect and dismantle possible evasion attempts. We tested our system against eXait, a tool containing a set of plugins that aim at detecting when a program is instrumented by Intel Pin, showing that Arancino is able to hide Intel Pin, allowing the analysis of evasive binaries.

Armed with Arancino, we then performed a large-scale measurement of the anti-instrumentation techniques employed by modern malware. We collected and analyzed 7,006 malware samples, monitoring the evasive behaviors that triggered our system, hence studying the common techniques adopted by modern malware authors to perform evasion of instrumentation systems.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Polino-Hiding-Pins-Artifacts-To-Defeat-Evasive-Malware.pdf)

Presented By

Mario Polino & Andrea Continella & Sebastiano Mariani & Lorenzo Fontana & Stefano D'Alessio & Fabio Gritti & Stefano Zanero

---

How Samsung Secures Your Wallet and How to Break It

Samsung launched its mobile payment service -- SamsungPay. For about two years, few discussion have been made public. In this talk, we will demystify every detail of SamsungPay implementation and security mechanism. In addition, we will discuss the way to bypass those obstacle that block analyzing.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Ma-How-Samsung-Secures-Your-Wallet-And-How-To-Break-It.pdf)

Presented By

HC MA

How to Hack a Turned-Off Computer or Running Unsigned Code in Intel Management Engine

Intel Management Engine is a proprietary technology that consists of a microcontroller integrated into the Platform Controller Hub (PCH) microchip with a set of built-in peripherals. The PCH carries almost all communication between the processor and external devices; therefore Intel ME has access to almost all data on the computer, and the ability to execute third-party code allows compromising the platform completely. Researchers have been long interested in such "God mode" capabilities, but recently we have seen a surge of interest in Intel ME. One of the reasons is the transition of this subsystem to a new hardware (x86) and software (modified MINIX as an operating system) architecture. The x86 platform allows researchers to bring to bear all the power of binary code analysis tools.

Unfortunately, this changing did not go without errors. In a subsystem change that will be detailed in the talk of Intel ME version 11+, a vulnerability was found. It allows an attacker of the machine to run unsigned code in PCH on any motherboard via Skylake+. The main system can remain functional, so the user may not even suspect that his or her computer now has malware resistant to reinstalling of the OS and updating BIOS. Running your own code on ME gives unlimited possibilities for researchers, because it allows exploring the system in dynamics.

In our presentation, we will tell how we detected and exploited the vulnerability, and bypassed built-in protection mechanisms.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Goryachy-How-To-Hack-A-Turned-Off-Computer-Or-Running-Unsigned-Code-In-Intel-Management-Engine.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Goryachy-How-To-Hack-A-Turned-Off-Computer-Or-Running-Unsigned-Code-In-Intel-Management-Engine-wp.pdf)

Presented By

Maxim Goryachy & Mark Ermolov

This talk will be 50% real audio from a social engineering engagement and 50% lessons learned from the call. During this call I talk a VP at a bank into giving us full access to his computer as well as facilities. At one point during the call, the AV triggers (thanks to a junior submitting the payload to virustotal :)). This is an intense call with a ton of valuable lessons for any social engineer or defender looking to learn how to identify attacks.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu17-Crumbaugh-How-To-Rob-A-Bank-Over-The-Phone.pdf)

Presented By

Joshua Crumbaugh

I Trust My Zombies: A Trust-Enabled Botnet

Defending against botnets has always been a cat and mouse game. Cyber-security researchers and government agencies attempt to detect and take down botnets by playing the role of the cat. In this context, a lot of work has been done towards reverse engineering certain variants of malware families as well as understanding the network protocols of botnets to identify their weaknesses (if any) and exploit them. While this is necessary, such an approach offers the botmasters the ability to quickly counteract the defenders by simply performing small changes in their arsenals.

We attempt a different approach by actually taking the role of the Botmaster, to eventually anticipate his behavior. That said, in this presentation, we present a novel computational trust mechanism for fully distributed botnets that allows for a resilient and stealthy management of the infected machines (zombies). We exploit the highly researched area of computational trust to create an autonomous mechanism that ensures the avoidance of common botnet tracking mechanisms such as sensors and crawlers. In our futuristic botnet, zombies are both smart and cautious. They are cautious in the sense that they are careful with whom they communicate with. Moreover, they are smart enough to learn from their experiences and infer whether their fellow zombies are indeed who they claim to be and not government agencies' spies. We study different computational trust models, mainly based on Bayesian inference, to evaluate their advantages and disadvantages in the context of a distributed botnet. Furthermore, we show, via our experimental results, that our approach is significantly stronger than any technique that has been seen in botnets to date. Finally, we step out of the adversarial perspective and touch the topic of countermeasures against our own approach.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Vasilomanolakis-I-Trust-My-Zombies-A-Trust-Enabled-Botnet.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Vasilomanolakis-I-Trust-My-Zombies-A-Trust-Enabled-Botnet-wp.pdf)

Presented By

Emmanouil Vasilomanolakis & Max Mühlhäuser & Jan Helge Wolf & Leon Böck & Shankar Karuppayah

---

Inside Android's SafetyNet Attestation

Many app developers often have questions like the following: "Is the device my app runs on reliable and trustworthy?" "Could it be, god forbid, 'rooted'?" It turns out that answering these

questions is quite difficult. In an area traditionally dominated by "root detection" products and DIY techniques, Google attempts to respond to this request: "OK Google, what do you think about the device I'm running in?"

SafetyNet is the primary security platform used by Google to keep the Android ecosystem in check. SafetyNet Attestation is a service offered by the SafetyNet system to all Android application developers, who can use it to gain some insight into what Google believes is the state of tampering of the operating system and the device.

Unfortunately, SafetyNet Attestation is not well documented by Google. How does it work? What checks does it do? Does it really help? How can you implement it in your app without it being trivially bypassable? Taking a perspective useful to both developers and penetration testers, this presentation covers multiple aspects of the system.

Part one of this presentation will quickly recap the basics of root detection and tamper detection on Android applications. Part two takes a deep dive into the internals of the SafetyNet system and Attestation specifically, what checks it does and how it is designed, detailing how it different to traditional detection techniques. Part three discusses the different ways the system can be implemented in real world applications and how each method may achieve different level of risk reduction. This is based on the lessons learned from implementing SafetyNet Attestation for several apps with large install bases and will show how an organization's maturity may impact security checks. Finally, part four presents various attacks and bypasses against SafetyNet Attestation which target not only SafetyNet but other similar approaches.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Mulliner-Inside-Androids-SafetyNet-Attestation.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Mulliner-Inside-Androids-SafetyNet-Attestation-wp.pdf)

Presented By

Collin Mulliner & John Kozyrakis

Intel ME: Flash File System Explained

Intel Management Engine (ME) technology has been around for over 10 years (since 2005), but it seems impossible to find any official information about ME on the Internet. Fortunately, some studies have been published in recent years; however, all of them deal with ME 10 and earlier, while modern computers implement ME 11 (introduced in 2015 for Skylake microarchitecture). In our presentation, we explain in detail how ME 11.x stores its state on the flash and the other types of file systems that are supported by ME 11.x.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Sklyarov-Intel-ME-Flash-File-System-Explained.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Sklyarov-Intel-ME-Flash-File-System-Explained-wp.pdf)

Presented By

Dmitry Sklyarov

Jailbreaking Apple Watch

On April 24, 2015, Apple launched themselves into the wearables category with the introduction of Apple Watch. This June, at Apple's Worldwide Developer Conference, Apple announced that their watch is not only the #1 selling smartwatch worldwide by far, but also announced the introduction of new capabilities that will come with the release of watchOS 4. Like other devices, Apple Watch can contain user data such as email and text messages, contacts, GPS and more, and like other devices and operating systems, has become a target for malicious activity.

In the Apple ecosystem, in order to explore the internals and security aspects of an Apple iOS based device it's necessary to use a jailbreak. However, a jailbreak does not exist publicly for watchOS so we had to create the first ever public Apple Watch jailbreak. This talk will take us inside the mind of a researcher, showcasing the unique set of skills, determination and rationalization needed from someone in order to piece this jailbreak together from scratch. We will provide an overview of Apple Watch and watchOS security mechanisms including codesign enforcement, sandboxing, memory protections and more. This will ultimately lead to a demonstration and explanation of the jailbreak and what we were able to learn about its general structure and ability to access iPhone-synced data.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Bazaliy-Jailbreaking-Apple-Watch.pdf)

Presented By

Max Bazaliy

Key Reinstallation Attacks: Breaking the WPA2 Protocol

We introduce key reinstallation attacks. These attacks abuse features of a protocol to reinstall an already in-use key, thereby resetting

nonces and/or replay counters associated to this key. We show that our novel attack technique breaks several handshakes that are used in a WPA2-protected network.

All protected Wi-Fi networks use the 4-way handshake to generate fresh session keys. The design of this handshake was proven secure, and over its 14-year lifetime no weaknesses have been found in it. However, contrary to this history, we show that the 4-way handshake is vulnerable to key reinstallation attacks. In such an attack, the adversary tricks a victim into reinstalling an already in-use key. This is achieved by manipulating and replaying handshake messages. When the victim reinstalls the key, the associated incremental nonce and replay counter is reset to its initial value. Apart from breaking the 4-way handshake, we also show that our key reinstallation attack breaks the group key and Fast BSS Transition (FT) handshake. The impact of our attacks depend on both the handshake being targeted, and the data-confidentiality protocol in use. Simplified, against AES-CCMP, an adversary can replay and decrypt packets, but cannot forge packets. Still, this makes it possible to hijack TCP streams and inject malicious data into them. Against WPA-TKIP

and GCMP, the impact is catastrophic: an adversary can replay, decrypt, and forge arbitrary packets. Rather surprisingly, GCMP is especially affected because it uses the same authentication key in both communication directions.

Finally, we confirmed our findings in practice, and found that every Wi-Fi device is vulnerable to some variant of our attacks. Notably, our attack is exceptionally devastating against Android and Linux: it forces the client into using a predictable all-zero encryption key.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Vanhoef-Key-Reinstallation-Attacks-Breaking-The-WPA2-Protocol.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Vanhoef-Key-Reinstallation-Attacks-Breaking-The-WPA2-Protocol-wp.pdf)

Presented By

Mathy Vanhoef

Locknote: Conclusions and Key Takeaways from Black Hat Europe 2017

At the close of this year's conference, join Black Hat Founder Jeff Moss and members of the esteemed Black Hat Review Board for an insightful conversation on the most pressing issues facing the InfoSec community. This Locknote will feature a candid discussion on the key takeaways coming out of Black Hat Europe and how these trends will impact future InfoSec strategies.

Presented By

Jeff Moss & Sharon Conheady & Andreas Lindh & Shawn Moyer

Lost in Transaction: Process Doppelgänging

Process Hollowing is a technique first introduced years ago by attackers to thwart the mitigation capabilities of security products. However, most of today's solutions are able to detect and prevent such notorious attacks. In this talk, we will present a new technique, dubbed Process Doppelgänging, which has similar advantages but is much harder to detect - let alone prevent. Moreover, we will expose inherent limitations in the implementations of modern AV/NGAV scanning engines.

Most modern evasion techniques rely on complex memory manipulation in order to avoid AV/NGAV scan engines. Instead, we wanted to take advantage of the implementation of the Windows loader, and abuse it to load our code, while keeping it away from the prying eyes of security products. Moreover, the code will never be saved to any file on disk, making it invisible to most recording tools such as modern EDRs.

Doppelgänging works by utilizing two key distinct features together to mask the loading of a modified executable. By using NTFS transactions, we make changes to an executable file that will never actually be committed to disk. We will then use undocumented implementation details of

the process loading mechanism to load our modified executable, but not before rolling back the changes we made to the executable. The result of this procedure is creating a process from the modified executable, while deployed security mechanisms in the dark.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Liberman-Lost-In-Transaction-Process-Doppelganging.pdf)

Presented By

Tal Liberman & Eugene Kogan

Nation-State Moneymule's Hunting Season – APT Attacks Targeting Financial Institutions

Lazarus, Bluenoroff, and Andariel are three notorious APT groups from North Korea infamous for deconstruction, cyber heist, and espionage attacks. From DarkSeoul to Sony Picture Entertainment breach, the groups conducted several operations that have received international public attention.

Starting in 2016, we have observed a significant change in the targets and motivation of the groups. While the groups have a long history of conducting cybercrime and cyber espionage attacks, their operations have become more aggressive and more focused on the cybercrime attacks targeting financial institutions. In February 2016, a series of attacks from Lazarus group - which leveraged the SWIFT banking network used to target Bangladesh banks - were revealed. Later in May, the global WannaCry ransomware attack was also linked back to the nation. However, these attacks were just the tip of the iceberg.

In this talk, we will disclose five recent operations conducted by the groups. These operations targeted banks in Europe and South Korea, an ATM company and Bitcoin exchange service provider. One of the operations involved another ransomware attack conducted before the WannaCry operation. We will introduce the malware, vulnerabilities, IOC and TTP discovered in these attacks. In addition, we will show how we revealed the black-market trading and Bitcoin transaction performed by the attackers. In the hope of making the world a safer place, we disclose this information to help financial institutions react to the substantial threat.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Shen-Nation-State Moneymules-Hunting-Season-APT-Attacks-Targeting-Financial-Institutions.pdf)

Presented By

Chi-en (Ashley) Shen-Shen.html) & Kyoung-ju Kwak & Min-Chang Jang

---

Passive Fingerprinting of HTTP/2 Clients

HTTP/2 is the second major version of the HTTP protocol. It changes the way HTTP is transferred "on the wire" by introducing a full binary protocol that is made up of TCP connections, streams, and frames, rather than a plain-text protocol. Such a fundamental change from HTTP/1.x to

HTTP/2, means that client-side and server-side implementations have to incorporate completely new code in order to support new HTTP/2 features. This introduces nuances in protocol implementations, which, in return, might be used to passively fingerprint web clients.

Our research is based on more than 10 million HTTP/2 connections from which we extracted fingerprints for over 40,000 unique user agents across hundreds of implementations.

Reference: http://akamai.me/2qWIqON - whitepaper published by Akamai's Threat-Research Team.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Shuster-Passive-Fingerprinting-Of-HTTP2-Clients.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Shuster-Passive-Fingerprinting-Of-HTTP2-Clients-wp.pdf)

Presented By

Elad Shuster & Ory Segal

Red Team Techniques for Evading Bypassing and Disabling MS Advanced Threat Protection and Advanced Threat Analytics

Windows Defender Advanced Threat Protection is now available for all Blue Teams to utilize within Windows 10 Enterprise and Server 2012/16, which includes detection of post breach tools, tactics and techniques commonly used by Red Teams, as well as behavior analytics. Combined with Microsoft Advanced Threat Analytics for user behavior analytics across the Domain, Red Teamers will soon face a significantly more challenging time maintaining stealth while performing internal recon, lateral movement, and privilege escalation in Windows 10/Active Directory environments.

This talk highlights challenges to red teams posed by Microsoft's new tools based on common hacking tools/techniques, and covers techniques which can be used to bypass, disable, or avoid high severity alerts within Windows Defender ATP and Microsoft ATA, as well as TTP used against mature organizations that may have additional controls in place such as Event Log Forwarding and Sysmon.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Thompson-Red-Team-Techniques-For-Evading-Bypassing-And-Disabling-MS-Advanced-Threat-Protection-And-Advanced-Threat-Analytics.pdf)

Presented By

Chris Thompson

Ro(o)tten Apples: Vulnerability Heaven in the iOS Sandbox

In modern days, no exploitation chain can be considered complete without a reliable privilege escalation vulnerability. This is why many security researchers spend a lot of their research time in

finding those vulnerabilities. Apple has set a new standard in iOS security by implementing many innovative techniques to prevent exploitation of PE vulnerabilities, however despite their continuous efforts some areas of iOS still remain more exposed than others to this kind of vulnerabilities. This presentation will shed a light on some critical areas in the iOS kernel, that have been proven to contain many privilege escalation vulnerabilities that can potentially affect hundred of millions of iOS devices. In this talk, we will overview these yet unexplored areas and present a chain of vulnerabilities, leading to a complete kernel privilege escalation exploit while bypassing all the latest kernel mitigations Apple introduced.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Donenfeld-Rooten-Apples-Vulnerability-Heaven-In-The-IOS-Sandbox.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Donenfeld-Rooten-Apples-Vulnerability-Heaven-In-The-IOS-Sandbox-wp.pdf)

Presented By

Adam Donenfeld

Self-Verifying Authentication – A Framework for Safer Integrations of Single-Sign-On Services

SSO (single-sign-on) services, such as those provided by Facebook, Google and Microsoft Azure, are integrated into tens of millions of apps, websites and cloud services, just like the front door lock for every home. However, the integration practice is very ad-hoc: on one hand, protocol documentation and usage guides of SSO libraries are written by experts, who are like experienced "locksmiths"; one the other hand, most app/website programmers are not "locksmiths", and inevitably fall into many pitfalls due to misunderstandings of such informal documentation. Security bugs in SSO integrations are continuously discovered in the field, which leave the front door of the cloud wide-open for attackers. SSO bugs are the primary example when the Cloud Security Alliance ranked API integration bugs as the No. 4 top security threat. They have become a familiar theme in major security conferences, including BlackHat USA 2016 and BlackHat Europe 2016.

We are working on an open-source project, called SVAuth, to provide every website with a safer SSO integration, supported by formal program verification. SVAuth is ready for real-world adoption: (1) it is language independent, so it works with web apps in any language, such as PHP, ASP.NET, Python; (2) the default solution requires only a drop-in installation of an executable, without any library integration effort; (3) a programmer can customize the default solution for his/her special requirement. The customized solution will enjoy the same correctness assurance as the default one; (4) the SVAuth framework can accommodate all SSO services.

The main innovation underlying SVAuth is a program verification technology called SVX (or Self-Verifying Execution). It turns every SSO-protocol execution into a process of proving its own logic correctness: every time when a "lock" is being opened (i.e., a user is signing in), a "locksmith" (i.e., a program verifier) is always watching to assert whether it is a logically-sound normal procedure or a lock-picking attempt. In other words, executing protocol code becomes inseparable from verifying it. SVX has two other attributes which are magical: (1) the runtime overhead for verification is near

zero; (2) the self-verifying capability only needs to be built once into abstract classes of a protocol, and all concrete implementations derived from the protocol will automatically inherit the capability. Thus, the one-time verification effort in the protocol level is scaled up to all concrete implementations.

In this talk, we will first show and explain a number of SSO bugs that we discovered. They pinpoint the natural gaps between the perspectives of a protocol designer, an SDK provider and a regular website programmer. None of them can be called a "stupid bug". Then, we explain how SVX performs code verification, as well as the architecture of the SVAuth code. Finally, we give demos about real-world web apps using SVAuth.

The talk is based on two published papers, but contains many new contents reflecting our latest development.

[1] Securing Multiparty Online Services via Certification of Symbolic Transactions. In IEEE Symposium on Security and Privacy (S&P;) 2015. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/CST.pdf [2] Self-Verifying Execution. In IEEE Cybersecurity Development Conference (SecDev) 2016. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/09/Self-Verifying-Execution.pdf

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Chen-Self-Verifying-Authentication-A-Framework-For-Safer-Integrations-Of-Single-Sign-On-Services.pdf)

Presented By

Shuo Chen & Shaz Qadeer & Matt McCutchen & Phuong Cao & Ravishankar Iyer

---

The Apple of Your EFI: An Updated Analysis of the State of Apple's EFI Security Support

Duo Labs conducted an extensive data analysis on the state of Apple's EFI security from two main perspectives. The first was analysing all EFI updates released by Apple since OS X 10.10.0 through macOS 10.12.6 to fully characterise the security support provided across different Mac models and OS versions. This also provided a baseline for the "expected state" Mac systems should be in - this defined as the state the user would expect their Mac's software and firmware to be in after running the available updates. The second was an analysis across over 73,000 real-world Mac systems to compare the actual state of their EFI against the expected state.

Our findings cover a range of anomalies and security issues with the security support provided by Apple for their EFI firmware. More worryingly, our analysis shows significant deviations in the real-world state of EFI firmware in Macs compared to the expected state, which causes us to suspect a more systemic issue causing the failure of new EFI firmware that is supposed to be automatically installed alongside an OS update.

In addition to the data analysis discussed above, our research also aims to shine a light on the mechanisms used to update Apple's EFI itself - discussing how Apple's EFI updater tools operate and the controls they have in place. These insights come from the binary analysis of the tools

themselves, we are confident that this has not been discussed in this great of detail anywhere else - until now.

Alongside our findings in the form of a technical paper, we are also releasing the tools and APIs to enable admins and end users to have far greater visibility into the state of the EFI firmware on their Apple systems and to understand the security implications that it may contain.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Smith-The-Apple-Of-Your-EFI-An-Updated-Analysis-Of-The-State-Of-Apples-EFI-Security-Support.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Smith-The-Apple-Of-Your-EFI-An-Updated-Analysis-Of-The-State-Of-Apples-EFI-Security-Support-wp.pdf)

Presented By

Rich Smith & Pepijn Bruienne

The Great Escapes of VMWare: A Retrospective Case Study of VMWare G2H Escape Vulnerabilities

Virtual machine escape is the process of breaking out of the virtual machine and interacting with the host operating system. VMWare recently fixed several bugs in their products that were allowing malicious code to escape sandbox. Some of these issues were exploited and reported during exploitation contest and while others reported individually by researchers. For very obvious reason details of this bugs are undisclosed. This paper presents a case study of VMWare VM escape vulnerabilities based on the analysis of different patches released by VMWare in recent past.

Looking at the advisories published by VMWare in the last few months, reveals that there are many surfaces, that are being targeted by security researchers. To summarize, the attack surfaces would be as follows:

A) RPC Request handler. B) EMF Handler. C) VMWare Graphics Implementation.

Talking about vulnerabilities fixed in VMWare RPC layer, we see several CVEs (CVE-2017-4901, CVE-2016-7461 etc.) fixing security issues in RPC layers. This talk will cover end to end RPC implementation in VMWare workstation. It will cover everything from VMWare Backdoor in guest OS to different RPC command handler in host OS. We will uncover some of these fixed bugs in VMWare RPC layer by performing binary diffing on VMWare Workstation binaries. This talk will also showcase some of the PoCs developed from different VMware workstation patches.

VMWare's EMF file handler is one of most popular attack surfaces, when it comes to guest to host escape. VMSA-2016-0014 fixed several security issues in EMF file handling mechanism. EMF format is composed of many EMR data structures. TPView.dll parses every EMR structure in EMF file. In VMware, COM1 port is used by Guest to interact with Host printing proxy. EMF files are spool file format used in printing by windows. When a printing EMF file request comes from Guest, in host TPView.dll render the printing page. The TPView.dll holds the actual code which parses the EMF file structures. In our talk, we will be diving deep into this attack surface & uncover some of the

vulnerabilities fixed in this area recently by performing binary diffing on VMWare work station binaries.

VMSA-2017-0006 resolved several security vulnerabilities in Workstation, Fusion graphics implementation which allows Guest to Host Escape. These vulnerabilities were mostly present in VMWare SVGA implementation. In this section of our talk we will cover implementation of VMWare virtual GPU through reverse engineering different guest components (vmx_fb.dll - VMware SVGA II Display Driver, vmx_svga.sys - VMware SVGA II Miniport) as well as host component (vmware-vmx.exe) where virtualize GPU code exist. The VMware virtual GPU provides several memory ranges which is used by Guest OS to communicate with the emulated device. These memory ranges are 2D frame buffer and FIFO Memory Queue. In FIFO memory queue, we write command that we want our GPU to process. The way VMWare handles and process these commands is error prone. This talk will uncover some of these bugs in SVGA command processing code and try to understand anatomy of issues by bin-diffing through VMWare binaries.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Mandal-The-Great-Escapes-Of-Vmware-A-Retrospective-Case-Study-Of-Vmware-G2H-Escape-Vulnerabilities.pdf)

Presented By

Debasish Mandal & Yakun Zhang

The Spear to Break the Security Wall of S7CommPlus

In the past few years, attacks against industrial control systems (ICS) have increased year over year. Stuxnet in 2010 exploited the insecurity of the S7Comm protocol, the communication protocol used between Siemens Simatic S7 PLCs to cause serious damage in nuclear power facilities. After the exposure of Stuxnet, Siemens has implemented some security reinforcements into the S7Comm protocol. The current S7CommPlus protocol implementing encryption has been used in S7-1200 V4.0 and above, as well as S7-1500, to prevent attackers from controlling and damaging the PLC devices.

Is the current S7CommPlus a real high security protocol? This talk will demonstrate a spear that can break the security wall of the S7CommPlus protocol. First, we use software like Wireshark to analyze the communications between the Siemens TIA Portal and PLC devices. Then, using reverse debugging software like WinDbg and IDA we can break the encryption in the S7CommPlus protocol. Finally, we write a MFC program which can control the start and the stop of the PLC, as well as value changes of PLC's digital and analog inputs & outputs.

Based on the research above, we present two security proposals at both code level and protocol level to improve the security of Siemens PLC devices.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Lei-The-Spear-To-Break -The-Security-Wall-Of-S7CommPlus.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Lei-The-Spear-To-Break -The-Security-Wall-Of-S7CommPlus-wp.pdf)

Presented By

Lei Cheng

Wi-Fi Direct to Hell: Attacking Wi-Fi Direct Protocol Implementations

Today Wi-Fi is everywhere and is by far the most widely used wireless networking protocol. During the last years, Wi-Fi security research was mainly focused on WPA/WPA2 security mechanisms. But modern Wi-Fi firmware's and drivers support several protocols that could be targeted by attackers. This is the case of Wi-Fi P2P, also known as Wi-Fi Direct. This protocol provides with the ability to discover nearby devices and connect directly to each other via Wi-Fi without an intermediate access point.

This talk will present an in-depth security analysis of Wi-Fi Direct protocol including an architectural overview, description of the discovery process, description of the connection process and a description of the frame formats. Additionally, we will use Android, HP Printers, and Samsung Smart TVs among others as an example of vulnerable implementations. At the end of the presentation, we will release PoC for the vulnerabilities and a tool for fingerprinting devices supporting Wi-Fi Direct protocol.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Blanco-WI-FI-Direct-To-Hell-Attacking-WI-FI-Direct-Protocol-Implementations.pdf) [[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Blanco-WI-FI-Direct-To-Hell-Attacking-WI-FI-Direct-Protocol-Implementations-wp.pdf)

Presented By

Andrés Blanco

Zero Days Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits

Zero-day vulnerabilities and their exploits are useful in offensive operations as well as in defensive and academic settings.

RAND obtained rare access to a dataset of information about more than 200 zero-day software vulnerabilities and their exploits - many of which are still publicly unknown. We analyzed these data to provide insights about the zero-day vulnerability research and exploit development industry; give information on what proportion of zero-day vulnerabilities are alive (publicly unknown), dead (publicly known), or somewhere in between; and establish some baseline metrics regarding the average lifespan of zero-day vulnerabilities (longevity), the likelihood of another party discovering a vulnerability within a given time period (collision rate), and the time and costs involved in developing an exploit for a zero-day vulnerability.

The RAND study is the first publicly available research to examine vulnerabilities and their fully-functional exploits that are still currently unknown to the public. The research establishes initial

baseline metrics that can augment conventional proxy examples and expert opinion, inform ongoing policy discussions, and complement current efforts to related to retention and disclosure of zero-day vulnerabilities and exploits.

This research can help inform software vendors, vulnerability researchers, and policymakers by illuminating the overlap between vulnerabilities found privately and publicly, highlighting the characteristics of these vulnerabilities, and providing a behind-the-scenes look at zero-day exploit development.

[[image: image]](https://blackhat.com/docs/eu-17/materials/eu-17-Ablon-Zero-Days-Thousands-Of-Nights-The-Life-And-Times-Of-Zero-Day-Vulnerabilities-And-Their-Exploits.pdf)

Presented By

Lillian Ablon
