Total surveillance made easy with VoIP phones

Total surveillance made easy with VoIP phones - .mario, gnucitizen.org.

- Published: date not stated
- Original: <https://www.gnucitizen.org/projects/total-surveillance-made-easy-with-voip-phones/>
- Preserved from: https://www.gnucitizen.org/projects/total-surveillance-made-easy-with-voip-phones/ (stored) on 2026-08-16
- Capture timestamp: 20081104074325
- 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.

[image: weâ€™ve been listening to you]

Remember the article about call jacking with the BT Home Hub? Here is something comparable but pretty new. Since Ronald and pdp had announced the router hacking challenge, I've decided to play around a little bit and as a result I've managed to find a rather interesting issue. Although not directly related to the router hacking contest, the results I've got were rather disturbing and made me get a totally new view on the VoIP phone security landscape.

Most modern VoIP phones come with a Web interface which you can use for administrative purposes mainly. The user can adjust the device's settings, like change the display messages, maintain the address books and on some models even make phone calls from the Web interface. Since the device we've tested supports that particular functionality (**call a number via a simple POST request**) I've decided to have a look at the markup a little bit closer, not knowing what hornet's nest I was eventually going to stir up. The device we are talking about is the **Snom 32x** - a pretty common device that can be found in several medium and large offices.

The Web interface of the Snom 32x VoIP phone suffers from many vulnerabilities that may cause financial and reputation trouble for the organizations that trusts them in their network. The vulnerabilities are based on common attack vectors, such as XSS and CSRF.

It is essential to keep in mind that, in order to attack the Snom 32x Web interface, the attacker requires knowledge of its IP address. This knowledge can be obtained by using a XSS Intranet scanning technique or via a plain guessing/bruteforce method. By looping through several IP addresses with JavaScript, looking for URLs like `http://xxx.xxx.xxx.xxx/img/logo2.gif` *, we can easily locate the Web interface of the device. Of course a quick nmap scan delivers equivalent results with more ease and less time required. It is important to note that some networks use predictable names which also can also be used to pin-point a vulnerable device.*

After discovering the device the attacker has several possibilities for evilness. Let's have a look at some of them:

- call arbitrary people via the Web interface via the means of CSRF.
- make the call history *invisible* by calling a number like `');` which causes the Flash application displaying the most recent calls to crash.

-

- steal the phone history from the logs including any other details attached to the calls via XHR.
- poison the address book with a persistent XSS - the name is encoded correctly but not the phone number.
- inject a JavaScript worm to gain total control over the user by changing the visible output by performing XHR-CSRF attack.
- change the settings of registered phones, including the displayed text on the phone's display.
- **Last but not least, monitoring the victim by making a phone call to the attacker's number, who in tern will accept the call and recording the incoming sound. Note that the phone doesn't give any noticeable feedback (ring tones, etc) while the victim was kept under surveillance. Keep in mind that the victim pays for the call.**

It is also important to note that:

- None of the tested forms use tokens or other methods to make it harder to perform CSRF attacks against the Web interface.
- The phone comes with a default password, which is 0000. You can pick this information from Google.

Let's have a look at a proper scenario. If the attacker knows the IP of the device's Web interface he/she can disable and/or hijack the whole phone within a few minutes. By crafting a XSS-CSRF vector he/she can inject a persistent XSS into the address book. When the victim visits the phone book, the XSS worm is silently executed and the attacker gains a total control over the interface and the actions that will be performed in the future. This also circumvents any protection mechanisms like VPN or comparable network layers, etc. **And yes, in the worst case scenario the attacker will be able to survey the sound in the room in and cleanup the logs afterwords.**

The whole device seemed to be especially designed for these issues to work. It is not like the situation where a small design flaw endangers the entire security model like many other cases. This time almost any available feature was buggy and vulnerable, and it kind of seems like a wonder that those phones are shipped all over the world and nothing major have happened yet. I've tried to patch the phone with the latest firmware but that didn't work - the phone was temporarily disabled after the process and when it began responding again the firmware version was still the same. So, even if there are patches for those devices they sometimes can't be used at all. It is very probable that other phones from other companies have the same or at least similar issues, so don't consider this post as a blame on a certain company. We'd rather point out severe issues that kind of compare to Web hacking scene in the nineties and have them fixed urgently - no matter what phone is used or which company have built it.

Please check the VoIP devices that you use and verify that the Web interfaces implements strong passwords and enforces some minimum security level. Contact the company who built the device and do bug them with you concerns. This will help generating a less dangerous situation for you and other VoIP phone users. Vulnerabilities like the one mentioned above could cost you or your company great amounts of money, reputation loss and even large scale information theft.

The proof of concept (POC) code can be found over here (please use it responsibly):

```
[http://www.gnucitizen.org/blog/total-surveillance-made-easy-with-voip-phones/snom.htm](http://www.gnucitizen.org/blog/t
```

... and some supporting screenshots follow here:

[[image: screen1]](http://www.gnucitizen.org/blog/total-surveillance-made-easy-with-voip-phones/screen1.png) [[image: screen2]](http://www.gnucitizen.org/blog/total-surveillance-made-easy-with-voip-phones/screen2.png) [[image: screen3]](http://www.gnucitizen.org/blog/total-surveillance-made-easy-with-voip-phones/screen3.png) [[image: screen4]] (http://www.gnucitizen.org/blog/total-surveillance-made-easy-with-voip-phones/screen4.png)

Snom has been contacted in several ways but we haven't got any responded yet. We will keep you posted about any news. Feel free to comment - we are very looking forward discussing those issues more in depth.

update: 14 Feb 2008

We've received a response from Snom. This is what they say.

> To prevent this we will take the following measurments: - We will publish an article on "how to make your snom phone saver" on our website (including a link to it on the start page) - We will send out a newsletter to all our registred VARS and distributers with this information - We will work on the FW to improve security (just checked, on FW Ver. 7 the Flash applet is disabled by default) - We will publish a new email adress, for security matters (mostlikly security@snom.com), which goes to a bunch of people (including me).

We are thankful for letting us know about your future plans.