# SOLVING THE (IN)SECURITY OF HOME NETWORKED DEVICES

*Martin Smarda & Pavel Sramek*
Avast Software, Czech Republic

Email {smarda, sramek}@avast.com

## ABSTRACT

In the past few years, not a *VB* conference has gone by without a talk about someone hacking the devices they have at home. Be it routers, NAS-es or 'smart' TVs, there has always been one thing in common: the vendors have ignored the problems and refused to patch their products.

We are developing an automated vulnerability scanner intended to test devices without our code running on them. The intention is to educate users about misconfigurations and vulnerabilities that are detectable from another device in the network. Integrating such a scanner into consumer AV brings home network security to a new level and increases user awareness of those issues. We will present the technology and describe the challenges we have encountered on the way towards accomplishing this goal via maximizing the impact of even the simplest vulnerability scans.

A single researcher reporting an issue is simply not enough pressure to affect manufacturers' decisions. But what if we could make millions of users report the problem to their vendor or start replacing their devices with more secure ones?

## MOTIVATION

The 'Internet of Things' (IoT) has become one of the hottest buzzwords in recent years. Common consumer electronics now encompass smartphones, smart TVs, networked data storages and printers. All of these communicate using the Internet Protocol (IP) and are interconnected by network elements like routers and modems. New IP-enabled devices continue to emerge every year, the latest being 'smart' thermostats and light bulbs.

In recent years, there hasn't been a *Virus Bulletin* conference without a talk on the vulnerability of these 'smart' devices, the most recent being the excellent talk at VB2014 by David Jacoby [1]. All of those talks have had one thing in common: manufacturers ignoring responsible disclosure and leaving their older devices unpatched and vulnerable. And the explanation is at hand: the lifetime of a new device introduced to the market is so short that performing security audits, or even supporting anything older than a month or so makes no economic sense when the next model is already in the pipeline for release.

There is no way that security researchers alone can change the status quo. The only force strong enough to affect the manufacturers' decisions is the mass of consumers actually paying for their devices. As long as consumers are unaware of these threats, nothing is going to change. However, should the security industry focus on informing users about the dire state of their devices, instead of just staying silently in the background, there is hope. In this paper, we will discuss the way we believe this could be achieved.

## OUR PROPOSAL

Some may argue that securing networks is already a trend, that there are various software products for network monitoring, intrusion detection/prevention and many penetration-testing agencies. But those solutions target large enterprises and their sysadmins, not end-users. Without technical expertise and the resources to hire expensive security experts, consumers are left in harm's way.

In 2014, there were several statements claiming that the anti-virus (AV) industry is dead [2]. From our perspective, this is just playing with words. Traditional plain file-based AV may no longer be enough for securing endpoint devices, but that's just a symptom of the shifting threat landscape. Computer security is now more relevant than ever and attackers use more vectors than just files on floppy disks. As the threat space grows, consumer AV has evolved into a complex security solution, protecting users on multiple fronts. Since attacks targeting vulnerable home non-PC devices have emerged, expanding the security solution to cover them as well is the logical next step.

We want to address the network security issues threatening home users and small businesses. And since staying quiet about vulnerable devices is, as explained, not going to cut it, our goal is to make users aware of the vulnerabilities of devices they are using. Whenever possible, an idealized network security software will help its users with fixing or mitigating any problems it identifies.

A pilot network security module has been a part of *Avast Antivirus* for about a year and thus spread across our vast user base. Since the line between malware and network threats is quite fuzzy (think DNS/PAC hijacks [3]), integrating a simple scanner made sense. We have since learned that home network security is indeed a hot area and is very likely to attract lots of cybercriminals in the near future, especially as devices with valuable data continue to fragment away from the once dominant *Windows* platform, but keep being served by the same vulnerable routers.

We hope that by increasing user awareness, we'll start applying at least a little bit of pressure to device manufacturers and vendors. Hopefully, this will ultimately lead to them maintaining their products better and for longer periods, ideally for at least as long as the warranty lasts and not just the timeframe until the release of a new model.

## VULNERABILITIES: CLASSIFICATION

Home and small business end-users are threatened by a variety of networked device vulnerabilities. These can be classified using several different criteria to help identify the ones that are most important to report or attempt to mitigate.

### Generic classification

Vulnerabilities can be split into two generic types, general and specific:

- General vulnerabilities common to many devices of the same kind (e.g. routers). Usually caused by using deprecated systems / services / libraries with known issues. These tend to be highly prevalent.
- Specific vulnerabilities affecting only limited amount of devices, typically a specific model or a family of model

made by one manufacturer. Their prevalence is usually lower.

### Exploitation impacts

- The most immediate threat currently posed to end-users by vulnerabilities in their devices is enabling man-in-the-middle traffic hijacking. We've already observed malware changing router DNS settings to point to malicious servers.
- Attacks on devices running vulnerable embedded *Linux* systems may seek similar goals to attacks targeting *Linux*-based servers, i.e. taking control of the system and forming a botnet, typically with spam or DDoS as primary use.
- Some vulnerabilities threaten users indirectly by allowing unauthenticated access to an otherwise secure device, for example by leaking admin credentials to an attacker. After gaining access, the attacker can use it for one of the above attacks. Vulnerabilities of this kind are fairly common – for example, our telemetry showed that even a full year after its large media exposure, the 'rom-0' vulnerability affected roughly 1.4% of all consumer systems that ran our scan at the start of May 2015, meaning that one in about 70 home routers is still open to this particular attack vector.
- We also predict that, in the future, ransomware and denial-of-service attacks that abuse vulnerabilities will take the device's useful value away from the owner. Given the way IoT devices operate, this would be too big an opportunity for cybercriminals to pass up.

### Ability to fix

- *De facto* vulnerabilities caused by weak settings are the easiest to detect and fix, or to give guidance for fixing (for example using WEP encryption for securing a Wi-Fi network).
- System-level vulnerabilities which require a firmware update to fix are a harder challenge. Even if the firmware update is available, the patching process usually requires at least moderate IT proficiency and willingness to cooperate. Neither can be relied upon.
- Vulnerabilities without an available patch are, by definition, unfixable. In some cases, there may be a workaround based on disabling a vulnerable service. However, this requires the user to do so as per the previous point, and also removes a potentially desired functionality of the device.

Our goal is not to cover every vulnerability under the sun, since the manpower requirements to do so would be prohibitive, and scanning for them is not free either. Instead, we have put in place a process to progressively select the most important vulnerabilities to focus on – maximize coverage relative to prevalence and potential impact for end-users.

## SCANNING FOR VULNERABILITIES

Another, not often mentioned, factor of fighting vulnerabilities is the challenge of creating a safe automated scanning program for checking whether or not devices are vulnerable. While the basic scheme is simple – performing some kind of request and judging the result from the device's response or a side channel – there are important implications to consider.

The most important aspect is the destructiveness of a vulnerability check. This, indirectly, is yet another way to classify vulnerabilities.

- Vulnerabilities with non-destructive scanning methods allow their presence on a device to repeatedly and directly be verified by normal query operations, without affecting the device's functionality in any way after the scanning request is made. Authentication bypass vulnerabilities typically fall into this category.
- Vulnerabilities with destructive scanning methods only allow their presence to be checked in unsafe ways that, with non-zero probability, render the device at least partially inoperable when the exploitation probe executes.
- Finally, there is a grey zone in between. This must be judged on a case-by-case basis. Leaking a service password over a secured wireless network with the user notified straight away is different from forcing a restart of one service by crashing it during the check, which in turn is different from rebooting the entire device. Where to draw the line is an open question.

A security solution is not a robotized penetration tester. It must be restricted to scannng directly with non-destructive methods, because even the slightest bit of network disruption is a completely unwanted effect, damaging the security suite's reputation. Of course, there is always indirect detection – if one knows that a certain model with firmware of a certain version is vulnerable, and positively identifies a device and its firmware version, declaring it vulnerable is completely legitimate even if no scan for the actual bug takes place on-site. However, positive device identification is not always possible.

Another rarely mentioned aspect of scanning for vulnerabilities is resource utilization, both concerning the resources of the device performing the scan and the devices being targeted. Most consumer tier network devices are built from the cheapest components that can handle the typical usage scenario. This may lead to increased response times for many concurrent requests and slow devices down considerably when they're being scanned. A scanning engine for worldwide deployment therefore has to be careful not to overload the network's weakest link. Parallel requests in particular are dangerous and must be used with caution. This, along with consideration for the total running time due to the load of the host machine, can severely limit the quality of certain vulnerability probes otherwise considered trivial. One such example is checking for weak and exploitable username-password pairs – this probe is effectively a dictionary attack, and with a short dictionary, some false negatives are inevitable. Ultimately, this is again a question of balancing convenience with accuracy, but with careful selection, a reasonable compromise is possible, while adaptive probes that scale with resource pool size are the most suitable answer.

Looking at the vulnerability scanning task from another angle, there is also the important decision as to whether a scan is actually desirable or not, given the type of network the

device is connected to. For the purpose of scanning engine behaviour, networks can be classified as:

- Home and small business networks, which constitute the primary use case – the users check and secure the devices they own, and scans should be run without restrictions.
- Public networks, such as those in airports, which are expected to be visited by consumers. Connecting to them increases threat exposure in some ways and makes certain scans more important while other scans become undesirable due to the user not being the owner of the infrastructure – imagine your AV telling you that your fast food restaurant's Wi-Fi router on 192.168.0.1 uses default admin/admin credentials...
- Enterprise networks, which aren't intended to be scanned by a consumer tool. These sometimes feature intrusion detection/prevention systems, and running a scan there might trigger false alarms, add work for the IT department, and if something is found anyway, the person that is alerted to the presence of a vulnerability will very likely not be the right person to handle such an alert.

Our proposal is primarily aimed at home and small business networks, with vulnerabilities of public networks being a secondary focus. Enterprise networks are better left to professional contractors. Therefore, an additional challenge arises in detecting, at least in typical cases, what kind of network the host system is connected to. There are often strong hints to follow in a simple heuristic approach, but ultimately a cloud-assisted classifier considering multiple network classification probe results over time would probably be a better solution.

The final challenge in scanning devices for vulnerabilities comes from the limitations of the host system itself. Since mobile platforms have network connectivity and security suite presence, they can legitimately host a network scanner, provided that its core is reasonably platform-independent. Their networking capabilities, however, may be limited or intentionally restricted in comparison to desktop PCs, or they may not allow reasonably fast executable code updates to account for new detection routines. Therefore, our proposal includes the possibility of operating in a platform-independent fashion, and supports a back-end deployment mode, where a restricted device only probes the network while processing takes place on a remote server. This way, we can squeeze the maximum detection capabilities out of every platform, leading to the maximum possible user awareness.

All in all, the above paragraphs emphasize the value of correctly selecting which vulnerabilities should be scanned for, how, and under which circumstances. Thus, by choosing the highest-impact, lowest-risk vulnerabilities with sensible mitigation recommendations, instead of going for overall count or strictly by prevalence, we can hope that a consumer AV feature will help start a global improvement.

## SMART DETECTION ENGINE

To address the problems outlined in the previous sections, it is necessary to make the vulnerability scanning engine 'smart' in several ways. The engine itself should be able to determine when, and if at all, a vulnerability detection routine should be executed, and which supplementary data-gathering scans to perform beforehand. It must also be easy to update with new definitions by virus lab staff without crumbling as new detection routines are added. When evaluating the requirements, we have decided to replace the simple engine of our current network security module with a scalable data-driven one.

Since both direct and indirect vulnerability detection is algorithmic by nature, as opposed to static data like hashes or signatures in traditional AV, the problem our engine actually solves is scheduling tasks with known prerequisites, using data obtained in real time during the scanning process. Such an arrangement not only allows just the relevant scans to be performed on a network instance, but also scores high on maintainability.

Authors of detection routines can focus on vulnerability selection and implementing actual detection code, rather than integration with engine logic. The detection routines for this engine are supposed to be created by virus analysts as part of their workflow, since they are the ones who get to see traditional malware starting to migrate to networked devices.

This is illustrated by a recently sighted piece of malware originally from Brazil, detected by *Avast* as one of the HTML:DNSChanger-* variants. Even from a quick glance at the malicious code (Figure 1), it is easy to tell that the malware is using a unique attack vector: a combination of CSRF exploitation on a compromised website and weak default credentials in home routers (on predictable IP addresses) to plant a malicious DNS server address to router settings - the DNS server, under attackers' control, then redirected traffic to another attacker-controlled server, performing a phishing attack to harvest user credentials for various sites. The second half of the chain is nothing new. A script embedded in a web page, attacking a vulnerable router, however, is not so run-of-the-mill. This early example shows that good maintainability by those in contact with real-world threats is a highly desired feature of any new engine – it keeps reaction time as short as possible.

```
<script>
function attack(){
        new
        Image().src='http://192.168.1.1/userRpm/PPPo
        ECfgAdvRpm.htm?wan=0&lcpMru=1480&ServiceName
        =&AcName=&EchoReq=0&manual=2&dnsserver=58.20
        .       &dnsserver2=58.20.      &downBandwid
        th=0&upBandwidth=0&Save=%B1%A3+%B4%E6&Advanc
        ed=Advanced';
}
</script>
<img
src="http://admin:admin@192.168.1.1/images/logo.jpg"
 onload="attack()">
```

*Figure 1: Even from a quick glance at the malicious code, it is easy to tell that the malware is using a unique attack vector: a combination of CSRF exploitation on a compromised website and weak default credentials in home routers.*

While smart operation and maintainability are important, there is one last requirement for the engine concept, and that is good scalability – adding new vulnerability definitions shouldn't hinder the engine's performance, and a network scan shouldn't be slower because of it, unless the target device for the new definition is actually present and being scanned. We believe that the data-driven approach satisfies

this condition nicely. It is by creating engines like this that we can turn network device security into a common component of consumer security suites, at least on par with a sandbox or a firewall.

## INTERACTION WITH USERS AND VENDORS, ETHICAL ISSUES

Once network security scanning becomes a common part of AV security suites and the general public becomes more aware of how dangerous vulnerable devices can be, the optimistic view is that demand for devices considered 'secure' will rise and vendors will be pressured into auditing their new devices and supporting older ones or risk their brand's reputation.

The path to such a utopian scenario contains many unanswered question that aren't technical in nature. Starting with the basic premise, let's say we have the engine and definitions ready and cover more than just a few vulnerabilities. A significant, two-digit percentage of users should be notified about the flaws in their devices. How do we do it? Informing users is a delicate but key aspect of the optimistic scenario. In quite a lot of vulnerability cases, there is not much our security products can do to 'magically' fix the issue, despite being expected to do so by people not educated in the matter. Some vulnerabilities can be mitigated, but require user cooperation to do so. Helping inexperienced users and communicating the facts in a clear and non-stressful manner is, for sure, a big job for UX experts.

If a vulnerability can either be neutralized by changing settings or completely removed by a firmware update, one of the obvious solutions is to guide users step-by-step through the device's interface to perform the necessary actions. This approach, however, is quite costly due to the effort of creating the step-by-step guides, updating them and, most importantly, translating them into most world languages (including separate screenshots for the different language settings of the device itself). Thus, guidance of this kind will probably remain limited to high-prevalence, high-impact vulnerabilities and common network-related security problems.

The other, more generic path towards helping users improve their security is to recommend they disable or replace the device. Taking it one step further to avoid a device being replaced by another one which is just as vulnerable, only recommendations for secure devices from trusted manufacturers should be presented. Naturally, defining who is trusted would require the introduction of a device security certification program and security update guarantees from the manufacturer. This, of course, is not going to happen on its own – vendors will not pay for a 'certification sticker' unless they see it as a competitive advantage, meaning that the 'I want secure devices' grassroots movement must already be present before the situation gets to this. Until then, recommendations would have to be made without a certification program, on only a reputation basis.

We've discussed the user interaction side of things, but there is also the vendor side. There are some big ethical questions here. Starting with the simple task of alerting users to flaws in their products – if done incorrectly, vendors may view it as a threat to their business and spend their resources fighting the security industry instead of patching vulnerabilities.

Additionally, how will they view detection routines? In the case of direct detection, those basically contain proof-of-concept code, available for anyone to copy. If a vulnerability has already been made public, then there is technically nothing wrong with this, but vendors might view it as damaging anyway due to the distribution scope, bringing the code to millions of machines and possibly alerting users about something that is easily abused, such as default passwords or reading secrets via path traversal. We certainly do not want a security suite to be used as a hacking or exploitation tool so we need to act carefully.

Finally, there is responsible vulnerability disclosure – once the boom is over and common vulnerabilities are accounted for in security suites, virus labs will probably move on to actual vulnerability discovery to keep up with 0-day equipped attackers. Will following standard disclosure guidelines be met with cooperation or with obstacles and lawsuits? And what about vulnerabilities that leak before the responsible disclosure process finishes? Put detection routines in place or wait for the deadline? Just as interacting with users was hard on UX, interacting with vendors is going to be hard on lawyers.

## CONCLUSION

The current state of security of small networks and the non-PC devices that constitute them is unsound and device vulnerabilities are starting to become targets for cybercriminals. *Avast* is developing a robust security solution to help users secure their networks and, hopefully, start changing the state for the better by making those who make the purchases aware of manufacturers' negligence.

We believe that by setting a trend and educating users, there will eventually be enough pressure on the manufacturers for making and supporting secure devices to become a competitive advantage, at which point the security situation would no longer be considered desperate. While that point is currently very far away, we do see a way to reach it eventually.

## ACKNOWLEDGEMENTS

We would like to thank all the teams behind the *Avast Home Network Security* solution. They have provided insights, ideas and expertise that made this paper what it is, and did a great job during the actual development.

Project lead: Lukáš Rypáček; Research team: Dmitriy Kuznetsov, Robert Žáček, Antonín Kříž; Windows development team; Mac development team: incl. Radek Brich; Virus lab: Antonín Hýža.

## REFERENCES

[1] Jacoby, D. How I hacked my own house! VB2014. https://www.virusbtn.com/conference/vb2014/abstracts/LM1-Jacoby.xml.

[2] Krebs, B. Antivirus is Dead: Long Live Antivirus! Krebs on Security. 2014 http://krebsonsecurity.com/2014/05/antivirus-is-dead-long-live-antivirus/.

[3] How to change your router DNS settings and avoid hijacking. 2014. https://blog.avast.com/2014/11/21/how-to-change-your-router-dns-settings-and-avoid-hijacking/.