Individual Submission
Internet Draft
Intended status: Informational
Expires: August 14, 2021

M. McFadden
internet policy advisors, ltd uk

February 14, 2021

Evolution of Endpoint Security - An Operational Perspective
draft-mcfadden-opsec-endp-evolve-00.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on August 14, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

This draft discusses the traditional model of security where endpoints in the network are protected by a variety of tools. It proposes a model for endpoints and then argues that the older, traditional approach is no longer sufficient for operational security at the endpoint. A series of operational examples are discussed in an Appendix.

Table of Contents

1. Introduction

There is currently significant discussion of the evolution of the Internet's treat model. The evolution of the Internet has brought new approaches to transport, connectivity, service provision and the type of devices connected to the Internet. This document is a discussion of the capabilities and limitations of security solutions based on the traditional strategy of protecting endpoint devices.

The typical approach of protecting endpoints is affected by the evolution of those endpoints and the emergence of threats that do not rely on exploits at the endpoints. The goal of this document is to provide an operational perspective on this evolution.

The focus here will be capabilities and limitations of endpoint-only security solutions and the impact of the evolution of the Internet's threat model on operational security.

Our goal with this review is to describe the benefits and limitations of endpoint security in the real world - from an operational perspective - rather than in the abstract. We aim to highlight security limitations that cannot be addressed by endpoint solutions and to suggest how these may be mitigated with the concept of a defence-in-depth approach, in order to increase the resilience against attacks and data breaches.

Finally, this draft suggests how this approach might affect protocol design.

2. Endpoints, Definition, Model and Scope

2.1. Scope

Endpoints are the origin and destination for a communication between parties. This encompasses User Equipment (UE) and the Host at the other end of the communication. This is a simplification of operational experience in the real world. In fact, there is an enormous variety of devices at the endpoints of communication of parties. However, a generalized approach to endpoints is not only possible, but the subject of other work in the IETF. For instance, the TEEP Working Group [1] has attempted to describe a generalized model for endpoints. As a companion to this work, a taxonomy of endpoints [2] is also provided.

The goal is to provide a uniform way to describe the security properties of endpoints in order to also describe the threat model, or attack surface, of those endpoints.

For example:

o The following would be considered UEs: a smartphone, a smart device, any IoT device, a laptop, a desktop, a workstation, etc.

o Hosts might be represented by the following examples: physical servers, virtual servers/machines, etc.

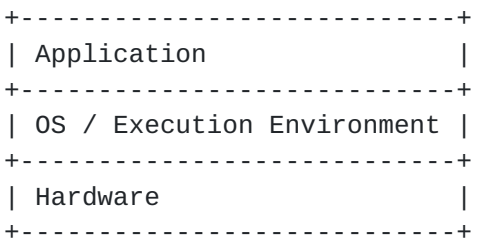
2.2. Models

In what follows, the discussion is limited to modeling the endpoints that are User Equipment (UE) rather than hosts. Having a model for Internet hosts is important but beyond the scope of this draft. The goal is to have a generalized description of the endpoint, its security properties, and the position of the endpoint in the network.

In addition, there are two separate descriptions for the endpoint. First, an internal view of the endpoint is provided and then a view of the external model for the endpoint is suggested. This approach provides a mechanism to cover the full attack surface and the threat landscape for the endpoint.

2.3. Internal representation of an endpoint

The companion endpoint taxonomy draft [2] provides a hierarchy of endpoint types, properties and capabilities. The taxonomy draft begins from the simplification represented in the following diagram:



Internal Endpoint Model

Today there are an enormous set of combinations of Hardware, OS/EE pairing, and Application layers, offering the user a vast set of features with a wide spectrum of capabilities. One of the features

of the evolution of the Internet's threat model is that the variety of these combinations is an enormous change compared to one or two decades ago.

The variety of combinations provide users with a rich set of new capabilities. Applications designers can provide services that were not possible even five years ago. However, with those new capabilities come new risks: new vectors for delivery of attacks and new attack surfaces at the endpoint.

The result is: the operational implications for endpoint security have evolved as the capabilities and variety of endpoints has expanded.

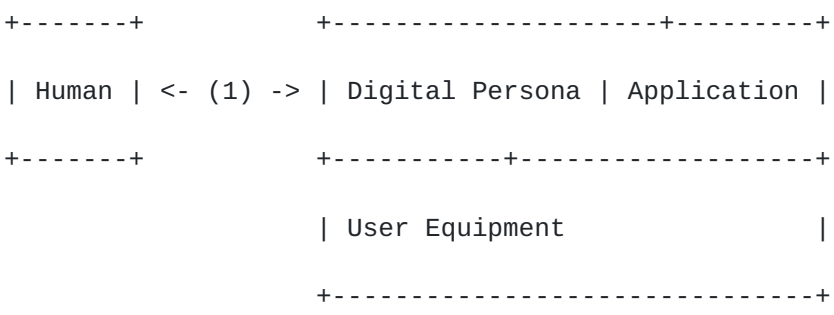
2.3.1. Applications as Endpoints

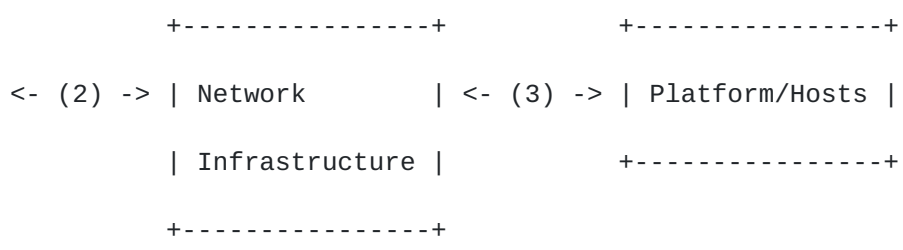
One of the other features of this evolution is that the application layer has evolved in such a way that it can be an endpoint as well. This has important operational security implications because prior approaches to providing endpoint security emphasized "per-device" security. In an evolved internet, the endpoint may be the application with multiple applications running on a single piece of User Equipment (UE). In this case, the threat model may be different for individual applications - resulting in different threat landscapes on a single device.

2.4. External representation of an endpoint

Section 2.3 describes a model for understanding the threat landscape of an individual UE device. In order to provide a complete view of the threat landscape there must also be a model for the connectivity properties of the device. This is a view of the attack surface of the device as viewed externally.

A representation of endpoints in an end-to-end context could look like the following diagram:





External Endpoint Model

1. Humans have a user experience (UX) with the UE, starting with an explicit or implicit Digital Persona, engaging with an application;

2. The application will have sessions through a network infrastructure. There are no assumptions made about how the network infrastructure is provisioned or what its properties are, (as an example, it could include landlines, mobile networks, satellites, etc.) nor are any assumptions made about the characteristics of those sessions (for instance, unicast or multicast, or sessions sourced from multiple hosts).

3. A platform consisting of many Hosts either physical or virtual which ensures a large part of the end-to-end user experience.

In this end-to-end model we see that many other systems may have interactions with the UE: the human, the UX, the digital persona, the sessions, the intermediate network infrastructure, and the hosts and applications at the destination.

This means that the operational security requirements for the model have expanded significantly. The threat landscape is very large and the attack surface will involve all the components and interactions at any level of either model.

3. Limitations of the Threat Landscape

In sections 2.3 and 2.4 above we saw that the pace of evolution of endpoints on the Internet has had a resulting change to the types and properties of attacks on those endpoints. In the past, the number of User Equipment (UE) was well-bounded and an approach to operational security that addressed the endpoints made intuitive and practical sense.

In today's Internet, the vast number and range of combinations that the generic modeling in sections 2.3 and 2.4 offers us, makes defining a threat landscape far more complicated. From an

operational perspective the prior approach of concentrating security on endpoints may have to be reconsidered.

3.1. Typical Categories of Threats

Any description of a threat model for endpoints must address typical, well-known attacks such as:

- Malware (Trojans, viruses, backdoors, bots, etc.)
- Adware and spyware
- Exploits
- Phishing
- Script-based attacks
- Ransomware, local Denial of Service (DoS) attacks
- Denial of Service (DoS) attacks
- Malicious removable storage devices (USB)
- In memory attacks
- Rootkits and firmware attacks
- Scams and online fraud
- System abuse (staging/proxying)

3.2. Evolution of threat types and their descriptions

However, the threat landscape is not static. Like the endpoints themselves, the threats evolve. It's difficult, for instance, to decide if cryptojacking or coinmining belong as examples of attacks in categories above. It is possible that they represent new categories of attack. In either case, a model of endpoint security must cope with a threat landscape that evolves as the properties of the endpoints evolve.

There are frameworks for describing threats:

O MITRE Common Attack Pattern Enumeration Classification (CAPEC). See [3]. CAPEC offers a hierarchical view of attack patterns by domains which can match some aspects of both of our above models,

but we will need to identify those attacks that fit exactly in our scope.

O MITRE ATT&CK. See [4]. ATT&CK offers a very straightforward categorized knowledge base of attacks, but it concentrates on the enterprise attack chain, so we will need to do some work to extract what we need.

In both cases, the frameworks do not address all of the threats that can affect the security of a system, for example they do not cover: routing hijacking, flooding, selective blocking, unauthorised modification of data sent to an endpoint, etc.

Phishing is an example of the limitations of using this approach. Phishing should be included as an attack, but while this is indeed an attack that will materialize on an end-device through an application (email, webmail, etc.), the real target of this attack is not the device, but the human behind the digital persona.

4. Endpoint Security Capabilities

For the purpose of this draft, endpoint security capabilities are the features that are used to protect the endpoint from attack. Protection has many meanings, however it is important to distinguish three different aspects of protection:

o Prevention - The attack doesn't succeed by intrinsic or explicit security capabilities.

o Detection - The attack is happening or has happened and is recorded and/or signaled to another component for action.

o Mitigation - Once detected, the attack can be halted or its effects can at least be reduced or reversed.

For example, prevention methods include keeping the software updated and patching vulnerabilities, implementing measures to authenticate the provenance of incoming data to stop the delivery of malicious content, or choosing strong passwords. Detection methods include inspecting logs or network traffic. Mitigation could include deploying backups to recover from an attack with minimal disruption.

The endpoint model definitions in sections 2.3 and 2.4 are simple but the security capabilities of each component may be complex. Each layer may or may not have a certain spectrum of intrinsic capabilities and there may be multiple ways to provide add-on and

third-party endpoint security capabilities, allowing complex interactions between all of these components.

4.1. Intrinsic versus added security

In terms of security capabilities for each layer of the model, there are those capabilities that are built-in or intrinsic, and there are those that are added to the layer through external means.

(A) Intrinsic security capability can be built-into each of the endpoint model layers

- (1) Hardware
- (2) OS/EE
- (3) Application

(B) Add-on security capability can be

- (4) a component of the hardware
- (5) a component of the OS/EE
- (6) an application by itself

In (A), there is a built-in, 'security by design' approach of the developer or manufacturers. They often offer a security model and security capabilities as part of their design.

In (B), a 3rd party is offering an additional security component which was not necessarily considered when the Hardware, OS/EE or Application was designed.

With regard to (6), there are many available options for add-on security capabilities offered by third-parties as applications on a commercial or open-source basis.

Gartner (see [5] highlights the evolution of endpoint security towards two directions as shown in [6], [7], and [8].

- Endpoint Protection Platform (EPP) as an integrated security solution designed to detect and block threats at the device level.
- Endpoint Detection and Response (EDR) as a combination of next generation tools to provide anomaly detection and alerting, forensic analysis and endpoint remediation capabilities.

4.2. Specific Endpoint Security Capabilities

Endpoint security capabilities can be grouped into four broad areas:

- those capabilities that are intrinsic to the endpoint and its network connectivity
- those that protect against the execution of problematic code
- those that protect against malware installation and execution
- those that protect against weaknesses in applications.

In the following sections, examples of these capabilities are provided.

4.2.1. Intrinsic Capabilities

- Software updates / patching
- Access Control (RBAC, ABAC, etc.)
- Authentication
- Authorization
- Detailed event logging

4.2.2. Execution protection

- Exploit mitigation (file/memory)
- Tamper protection
- Whitelisting filter by signatures, signed code or other means
- System hardening and lockdown (HIPS, trusted boot, etc.)

4.2.3. Malware protection

- Scanning - on access/on write/scheduled/quick scan (file/memory)
- Reputation-based blocking on files or by ML
- Behavior-based detection - (heuristic based/ML)
- Rootkit and firmware detection

- Threat intelligence based detection (cloud-based/on premise)

- Static detection - generic, by emulation, by ML, by signature

4.2.4. Attack/Exploit/Application Protection

4.2.4.1. Application protection (browser, messaging clients, social media, etc.)

- Disinformation Protection (anti-phishing, fake news, anti-spam, etc.)

- Detection of unintended link location (URL blocklist, etc.)

- Memory exploit mitigation, e.g. browsers

4.2.4.2. Network Protection (local firewall, IDS, IPS and local proxy) inbound and outbound

- Detection of network manipulation (ARP, DNS, etc.)

- Data Loss Prevention and exfiltration detection (incl. covert channels)

5. Optimal Properties of an Endpoint Security Solution

It is impossible to provide a complete list of the properties of an optimal endpoint security solution. The evolution of the threat landscape ensures that endpoint security solutions will require new features in the future. However, it is possible to provide a (non-exhaustive) list of properties - in the spirit of best practices - for an endpoint security solution.

- find instantly accurate reputation for any file before it gets executed and block it if needed.

- monitor any behavior on the endpoint, including inbound and outbound network traffic, learn and identify normal behavior and detect and block malicious actions, even if the attack is misusing legitimate clean system tools or hiding with a rootkit.

- patch instantly across all devices/systems/OSes, including virtual patching, meaning you can patch or shield an application even before an official patch is released.

- exploit protection methods for all processes where applicable, e.g. no execute bit (NX), data execution prevention (DEP), address

space layout randomization (ASLR), Control Flow Integrity Guard (CFI/CFG), stack canaries, shadow stack, reuse attack protection (RAP), etc. all of which are methods, which make it very difficult to successfully run any exploit, even for zero day vulnerabilities.

o detect attempts to re-route data to addresses other than those which the user intended, e.g. detect incorrectly served DNS entries, TLS connections to sites with invalid certificates, data that is being proxied without explicit user consent, etc.

o have an emulator/sandbox/micro virtualization to execute code and analyse the outcome and perform a roll back of all actions if needed, e.g. for ransomware.

o allow the endpoint to communicate with the other endpoints in the local network and globally, to learn from 'the crowd' and dynamically update rules based on its findings.

o be in constant sync with all other endpoints deployed on a network and other security solutions, run on any OS, with no delay (including offline modes and on legacy systems).

o run from the OS/EE when possible.

o run as one of the first process on the OS/EE and protect itself from any form of unwanted tampering.

o offers a reliable logging that can't be tampered with, even in the event of system compromise.

o receive updates instantly from a trusted central entity.

6. Case Studies in the Limitations of Endpoint Security Only

The previous section discusses what some of the optimal features of endpoint security 'system' would be. However, a more realistic view, based on operational security data would indicate:

o may not be able to run at full capacity due to computational power limits, battery life, performance, or policies (such as BYOD restrictions in enterprise networks), etc.

o may not be able to run at full capacity as it slows down performance too much.

o will miss some of the malware or attacks, regardless of detection method used, like signatures, heuristics, machine learning (ML), artificial intelligence (AI), etc.

o have some level of False Positives (FP).

o not monitoring or logging all activities on the system, e.g. due to constraints of disk space or when a clean windows tool is being triggered to do something malicious but the activity is not logged. Such activity can be logged, but a decision needs to be made if it's clean or not.

o have its own vulnerabilities or simple instabilities that could be used to compromise the system.

o be tampered with by the user, e.g. disabled or reconfigured.

o be tampered with by the attacker, e.g. exceptions added or log files wiped.

In the following section, a number of these limitations are reviewed in case studies.. Some limitations are absolute, and some limitations result in a grey area or suboptimality for the endpoint security solution.

6.1. Unable to put an endpoint security add-on on the UE

We have seen that UEs will vary a lot; by 2022, an estimated 29 billion devices will be connected, with 18 billion of them related to IoT [9]. Many IoT products lack the capacity to install any endpoint security capabilities, are unable to update the software, and it is not possible to force the UE provider to improve or even offer an intrinsic security capability.

In IoT we find UEs such as medical devices which are limited by regulation, welding robots that can't be slowed down, smart light bulbs which are limited by the processing power, etc. There are many factors influencing whether endpoint security can be added to a UE:

* The UE is simply not powerful enough or the performance hit is too high.

* Adding your own security will breach the warranty or will invalidate a certification or a regulation (breach of validity).

* The UE needs to run in real-time and any delay introduced by a security process might break the process.

* Some UEs are simply locked by design and the manufacturer does not provide a security solution (e.g. smart TV, fitness tracker or personal artificial assistants) see [10], [11].

In the future, a possible research problem would be to find hard data on the exact proportion of IoT devices that are unable to run any endpoint security add-on or that have no intrinsic security built-in.

The other hidden dimension here is the economical aspect. Many manufacturer are reluctant to invest in IoT device security, because it can significantly increases the cost of their solution and there is the perception that they will lose market shares, as customers are not prepared to pay the extra cost for added security.

6.1.1. Not receiving any updates or functioning patches

The endpoint security system may lack a built-in capability to be patched or it may be connected to a network that prevents the process of downloading updates automatically. For example stand-alone medical systems or industrial systems in isolated network segments often do not have a communication channel to the Internet.

Even if security updates are received, they typically will only be periodically updated; hence there will be a window of opportunity for an attacker, between the time the attack is first used, and the time the attack is discovered/patched and the patch is deployed.

In addition updates and patches may themselves be malicious by mistake, or on purpose if not properly authenticated, or if the source of the updates has malicious intent. This could be part of a software update supply chain attack or an elaborate attacker breaking the update process, as for example seen with the Flamer group (see [12]).

A recent survey found that fewer than 10% of consumer IoT companies follow vulnerability disclosure guidelines at all, which is regarded as a basic first step in patching vulnerabilities (see [13]). This indicates that many IoT devices do not have a defined update process or may not even create patches for most of the vulnerabilities.

Furthermore some endpoints system may reach the end of their support period and therefore no longer receive any updates for the OS/EE or the security solution due to missing licenses. However the systems may remain in use and become increasingly vulnerable as time goes on and new attacks are discovered.

In the following sections, individual examples of attacks on endpoints unable to provide for their own security are examined. In each case, the description of the attack follows the format of . . .

6.1.2. Mirai IoT bot

6.1.2.1. Mirai Description 1

| Description | A Mirai bot infecting various IoT devices through weak passwords over Telnet port TCP 23 and by using various vulnerabilities, for example the SonicWall GMS XML-RPC Remote Code Execution Vulnerability (CVE-2018-9866) on TCP port 21009. Once a device is compromised it will scan for further victims and then start a DoS attack. |

| Simplified attack process | Compromised device scans network for multiple open ports, attempts infection through weak password and exploits, downloads more payload, starts DoS attack. |

| UE | No security tool present on majority of IoT devices, hence no detection possible. If a rudimentary security solution with limited capabilities such as outgoing firewall is present on the IoT device e.g. router, then it might be able to detect the outbound DoS attack and slow it down. |

| References | [14] }[15] |

6.1.2.2. Mirai Description 2

| Name | Mirai |

| Description | A device infection for participation into a botnet activity |

| Endpoint Targeted | IoT Devices |

| Attack Surface Categories Involved | Telnet remote access; Weak default and existing passwords; Code vulnerabilities in exposed services |

| Attack Surface Examples | Weak passwords over Telnet TCP port 23; SonicWall GMS XML-RPC Remote Code Execution Vulnerability (CVE-2018-9866) on TCP port 21009 |

| Attack Objective | Deployment of a custom code or commands on the device for participation in botnet activities |

| Attack Category | Botnet Deployment; DDoS |

| Attack Orchestration | Exploit remote access weaknesses on the device to deploy a bot on the device |

| Mitigation | If a rudimentary security solution with limited capabilities such as outgoing firewall is present on the IoT device e.g. router, then it might be able to detect the added bot or the outbound DoS attack and slow it down |

| Attack Surface Minimisation | Better password management; Uptodate patching |

| References | [14] [15] |

6.2. Endpoints may not see the malware on the endpoint

6.2.1. LoJax UEFI rootkit

| Description | A device compromised with the LoJax UEFI rootkit, which is active before the OS/EE is started, hence before the endpoint security is active. It can pass back a clean 'image' when the security solution tries to scan the UEFI. Infection can either happen offline with physical access or through a dropper malware from the OS/EE. |

| UE | A perfect endpoint security could potentially detect the installation process if it is done from the OS/EE and not with physical modification or in the factory. Once the device is compromised the endpoint security solution can neither detect nor remove the rootkit. The endpoint solution may detect any of the exhibited behaviour, for example if the rootkit drops another malware onto the OS/EE at a later stage. |

| Reference | [16] |

6.2.2. SGX Malware

| Description | Malware can hide in the Intel Software Guard eXtensions (SGX) enclave chip feature. This is a hardware-isolated section of the CPU's processing memory. Code running inside the SGX can use return-oriented programming (ROP) to perform malicious actions. |

| UE | Since the SGX feature is by design out of reach for the OS/EE, an endpoint security solution can neither detect nor remove any injected malware. A perfect endpoint security solution could

potentially detect the installation process if it is done from the OS/EE and not with physical modification or in the factory. |

| References | [17] [18] |

6.2.3. AMT Takeover

| Description | A targeted attack group can remotely execute code on a system through the Intel AMT (Active Management Technology) vulnerability (CVE-2017-5689) over TCP ports 16992/16993. This provides full access to the computer, including remote keyboard and monitor access. The attacker can install malware, modify the system or steal information. |

| UE | The AMT is accessible even if the PC is turned off. Therefore any endpoint security software installed on the OS, would not be able to see this traffic and therefore also not able to detect it. |

| References | [19], [20] |

6.2.4. AMT case study (anonymised)

An enterprise has a data center containing very sensitive data. Their workstations use a certain Intel chipset which integrates the AMT feature for remote computer maintenance. AMT is an interface for hardware management of the workstations, including transmission of screen content and keyboard and mouse input for remote maintenance. Communication with the management workstation is implemented by AMT through the network interface card (NIC) on the motherboard. The network packets generated in this way are invisible both to the main processor and thus to the OS running on the workstation. In autumn of 2015, it became known that some AMT-enabled computers had a flaw that allowed AMT's remote maintenance component to be activated and configured by attackers. This also worked when the workstations were switched off. The leakage of data through this vulnerability is elusive and difficult to detect. The identified threat situation led the organization to a new requirement implementing a method that can reliably detect this and similar vulnerabilities. In particular, the detection of rootkits and manipulated firmware, and this includes also (UEFI) BIOS - has also been a focus of their attention.

The method used as a solution, compares the desired data packets generated by a client operating system - the user, with the data packets received on the switch port. If more data has been received on the switch port than was been sent by the operating system - the user, there is a strong possibility that something bad is happening - like for example an infection via modified firmware or by rootkit.

6.2.5. Users bypass the endpoint security

| Description | Endpoint security systems should not interfere with the normal operation of the endpoint to the extent that users become frustrated and want to disable them or configure them to disable a significant fraction of important security capabilities. |

| UE | Add-on endpoint security is now bypassed or disabled by the user. Unless the endpoint is under monitored management or can prevent a user from modifying the configuration, then this is shutting down a significant fraction of the security capabilities. |

| References | [21] |

6.3. Endpoints may miss information leakage attacks

Another aspect that endpoint security has issues in detecting are information disclosure or leakage attacks, especially on shared virtual/physical systems.

6.3.1. Meltdown/Specter

The Meltdown/Specter vulnerabilities and all its variants may allow reading of physical memory belonging to another virtual machine (VM) on the same physical system. This could reveal passwords, credentials, certificates etc. The trick is that an attacker can spin up his own VM on the same physical hardware. As this VM is controlled by the attacker, they will ensure that there is no endpoint security that detects the Meltdown exploit code when run. It is very difficult for the attacked VM to detect the memory read-outs. For known CPU vulnerabilities there are software patches available than can be applied. If it is an external service provider, it might not be in the power of the user to patch the physical system or to determine if this has been done by the provider.

6.3.2. Network daemon exploits

Other attack types, which leak memory data from a vulnerable web server, are quite difficult to detect for an endpoint security. For example the Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This could lead to credentials or keys being exposed. An endpoint solution needs to either patch the vulnerable application or monitor it for any signs of exploitation or data leakage and prevent the data from being exfiltrated.

6.3.3. SQL injection attacks

A SQL injection attack is an example of an attack that exploits the backend logic of an application. Typically, this is a web application with access to a database. By encoding specific command characters into the query string, additional SQL commands can be triggered. A successful attack can lead to the content of the whole database being exposed to the attacker. There are other similar attacks that can be grouped together for the purpose of this task, such as command injection or cross site scripting (XSS). Although they are different attacks, they all at their core fail at input filtering and validation, leading to unwanted actions being performed.

Applications that are vulnerable to SQL injections are very common and are not restricted to web applications. An endpoint solution needs to monitor all data entered into possible vulnerable applications. This should include data received from the network. A generic pattern matching for standard SQL injection attack strings can be applied to potentially block some of the attacks. In order to block all types of SQL injection attacks the endpoint solution should have some knowledge about the logic of the monitored application, which helps to determine how normal requests differ from attacks.

Applications can be analysed at source code level for potential weaknesses, but dynamically patching is very difficult. See { [22] }

6.3.4. Low and slow data exfiltration

An endpoint security solution can detect low and slow data exfiltration, for example when interesting data sources are tracked and access to them is monitored. If the data source is not on the endpoint itself, e.g. a database in the network, then the received data needs to be tagged and its further use needs to be tracked. To make detection difficult, an attacker could decide to use an exfiltration process that sends only 10 bytes every Sunday to a legitimate cloud service. If that is not in the normal behavior pattern, then this anomaly could be detected by the endpoint. If the process that sends the data or the destination IP address have a bad reputation, then they could be stopped. Though it is very difficult to reliably block such an attack and most solutions have a specific threshold that needs to be exceeded before it is detected as an anomaly.

6.4. Suboptimality and gray areas

6.4.1. Stolen credentials

Stolen credentials and misuse of system tools such as RDP, Telnet or SSH are a valid scenario during attacks. An attacker can use stolen credentials to remotely log into a system and access data or execute commands in this context like the legitimate user might do. An endpoint security solution can restrict access from specific IP addresses, but this is difficult in a dynamic environment and when an attacker might have already compromised a trusted device and misuse it as a stepping stone for lateral movement. The endpoint could perform additional checks of the source device, such as verifying installed applications and certain conditions. Again this will not work in all scenarios, e.g. a hijacked valid device during lateral movement.

This means that the system will not be able to simply block the connection if the authentication with the stolen credentials succeeds. A multi factor authentication (MFA) could limit the use of stolen credentials, but depending on the system used and the determination of the attacker they might be able to bypass this hurdle as well e.g. cloning a SIM card to read text message codes.

As a next step, a solution on the endpoint can monitor the behavior of the logged in user and determine if it represents expected normal behavior. Unfortunately, there is the chance for false positives that might block legitimate actions, hence the rules are usually not applied too tightly. The system can monitor for suspicious behavior, similar to malware detection, where every action is carefully analyzed and all activity is tracked. For example if the SSH user is adding all files to archives with passwords and then deletes the original files in the file explorer, then this could result in a ransomware case scenario. If only a few files are processed per hour, then this activity will be very difficult for the endpoint to distinguish from normal activity, in order to flag it as malicious.

The problem of attackers blending in with normal activity is one of the biggest challenges with so called living off the land attack methods. The attacker chooses to keep their profile low by not installing any additional binary files on the system, but instead misuses legitimate system tools to carry out their malicious intent. This means that there is no malware file that could be identified and the detection relies solely on other methods such as behaviour based monitoring { [23] }.

If information is shared across multiple endpoints, then each one could learn from the others and see how many connections came in from that source, what files were involved and what behavior the clients exhibited. This crowd wisdom approach would allow blocking rules to be applied after the first incident across multiple endpoints.

6.4.2. Zero Day Vulnerability

| Description | An attacker exploits a zero day vulnerability or any recent vulnerability. |

| UE | In theory this scenario could be handled by the endpoint security: a) Once the intrinsic security system has been patched, exploitation of the vulnerability can be prevented. b) The add-on security with enhanced capabilities or updated methods can detect and mitigate the vulnerability. It does not necessarily require the official patch.|

| Challenge | In practice many systems remain vulnerable to a vulnerability months or even years after a security fix has been released. Moreover there is a big gap between when a vulnerability is disclosed and when a security fix is available. Also there is a big gap between when a security fix is available and when the security fix is actually applied. A recent study over three years, examined the patching time of 12 client-side and 112 server-side applications in enterprise hosts and servers. It took over 6 months on average to patch 90% of the population across all vulnerabilities. { [24] }. We note too: "The patching of servers is overall much worse than the patching of client applications. On average a server application remains vulnerable for 7.5 months." |

| References | { [25] }{{ [26] } |

6.4.3. Port scan over the network

An infected machine, let's say a Mirai bot on a router, is scanning a class B network for IP addresses with TCP port 80 open. The malware can slow it down to 1 IP address per 5 seconds (or any other threshold) and it can go in randomized order (like for example the nmap tool does) in order to make it difficult to find a sequential pattern. To increase detection difficulties, legitimate requests to existing web servers can be added in at random intervals.

An endpoint solution might be able to detect this behaviour, depending on the threshold, but it will be difficult. At some point the pattern will be similar to browsing the web, so either the endpoint blocks the bot scanning and also the user from surfing, or it allows both.

To make it even harder, the attacker can use a botnet that communicates over peer-to-peer (P2P) or a central command and control server (C&C) and then distribute the scan load over multiple hosts. This means each endpoint only scans a subset, let's say 100 IP addresses, but all 1,000 bots scan a total of 100,000 IP addresses.

This attack is difficult to detect by a reasonable threshold on each endpoint individually. If the endpoints talk to each other and exchange information, then a collective decision can be made on the bigger picture of the bot traffic.

Another option for the endpoint solution is to block the bot malware from operating on the computer, for example by detecting the installation, analyzing the behavior of the process or by preventing the binary from accessing the network. This includes blocking any form of communication for the process to its C&C server, regardless of if it is using a P2P network or misusing legitimate system tools or browsers to communicate with the Internet. Blocking indirect communication over system tools as part of living off the land tactics, can be very challenging.

See { [27] }

6.4.4. DDoS attacks

For this example let us consider a botnet of 100,000 compromised computers and each one sends a burst of traffic to a remote target, for one second each, alternating in groups. This will generate some waves of pulse attack traffic. Similar comments can be made about overall pulsed DDoS attacks { [28] }.

A solution on the endpoint can attempt to detect the outgoing traffic. If the DoS attack is volume based and the time span of each pulse is large enough or the repeating frequency for each bot is high, then detection with thresholds on the endpoint is feasible. It is different, if it is an application layer DoS attack, where the logic of the receiving application is targeted, for example with too many search queries in HTTP GET requests. This would flood the backend server with intensive search requests, which can result in the web site no longer being responsive. Such attacks can succeed

with a low amount of requests being sent, especially if its distributed over a botnet. This makes it very difficult for a single endpoint to detect such an ongoing attack, without knowledge from other endpoints or the network.

Another option for the endpoint solution is to block the bot malware from operating on the computer, for example by detection the installation, analyzing the behavior of the process or by preventing the binary from accessing the network. This includes blocking any form of communication for the process to its C&C server, regardless of if it is using a P2P network or misusing legitimate system tools or browsers to communicate with the Internet. Blocking indirect communication over system tools as part of living off the land tactics, can be very challenging.

7. Defense-in-depth from the perspective of protocol design

While endpoint security systems have good capabilities (for instance, those seen in section 5 above), sometimes it is debatable and perhaps suboptimal to let the endpoint run the capability alone or at all. It is generally considered good security practice to adopt a defense-in-depth approach (see [29]). The Open Web Application Security Project group (OWASP) describes the concept as follows:

"The principle of defense-in-depth is that layered security mechanisms increase security of the system as a whole. If an attack causes one security mechanism to fail, other mechanisms may still provide the necessary security to protect the system." (see [29]).

Indeed, there are many other constituencies as per our end-to-end model that can participate in the defence process: The network, the infrastructure itself, the platform, the human, the user experience and in a hybrid of an on premise and cloud approach, an Integrated Cyber Defence (ICD) of the entire chain.

The simple idea behind the concept is that "every little bit helps". If the endpoint is not 100% secure itself, the detection chance can increase with additional security capabilities from other entities. We acknowledge that there are some case where adding an additional component to the system may degrade the overall security level by introducing new weaknesses.

There are various reference articles in the industry highlighting limitations of endpoint only solutions. For example this quote here, which talks about multi-tier solutions:

"There are limitations with any endpoint protection solution, however, that can limit protection to only the client layer. There is also a need for security above the client layer, as endpoint protection products cannot intercept traffic. Vendors will often sell a multi-tiered solution that enables a network appliance to assist the endpoint protection client by intercepting traffic between the attacker and the infected client. Vendors will also sell solutions that monitor and intercept traffic on internal or external network segments to protect the enterprise from these threats. A prime example of the limitations of endpoint protection software is infection via a phishing attack." [30].

Some sources point out that even the best solution might not get deployed in the optimal way in a real world scenario as the environment can be very complex:

"While endpoint security has improved significantly with the introduction of application whitelisting and other technologies, our systems and devices are simply too diverse and too interconnected to ensure that host security can be deployed 100% ubiquitously and 100% effectively." [31]

On these grounds it is considered a good idea to follow a layered approach when it comes to security.

"In today's complex threat environment, companies need to adopt a comprehensive, layered approach to security, which is a challenging task in such as rapidly evolving, crowded market." [331]

It is important to comprehend the capabilities of endpoint security solutions in this overall picture of the connected environment, which includes other systems, networks and various protocols that are used to interact with these entities. Understanding possible shortcomings from single layered solutions can help counterbalance such weaknesses in the architectural concept or the protocol design.

8. Endpoint security from the perspective of protocol design

The previous sections have reviewed two models for endpoints on the Internet, a review of how endpoints have evolved, an examination of the features of endpoint security and - finally - a view of how endpoint security, alone, may not be sufficient to protect endpoints and users in a evolved Internet. A taxonomy of endpoints is provided in a companion draft [2]. The evolution has brought a significant set of new threats and there is evidence that endpoint security is not enough to protect against those threats.

This is an observation about operational security, but what does it mean for protocol design?

8.1. Simplicity of design is important

Protocol design is often informed by attempting to get all possible use cases addressed in early versions of the protocol design. It is widely observed that subsequent versions of a protocol - popular or not - are difficult to achieve. However, simplicity in protocol design ensures that a protocol will be well-understood at design-time, it can be modelled and receive a full (and, perhaps formal) security analysis. The Security Directorate already routinely reviews protocols in Last Call for their security characteristics - a practice that should endure. However, making protocol design decisions that allow for easy, open review helps build confidence in the security of the underlying design.

It is certain that there will always be unintended consequences in choices made at protocol design time. However, if the underlying protocol is not overly complex, those unintended consequences can be limited. It also make possible modularization or code-reuse allowing for the reuse of security analysis from one protocol to another.

It is worth noting that implementation is also affected by protocol design decisions. Complex or poorly defined methods for deployment of software that implements a protocol means that they are more likely to be deployed incorrectly or insecurely.

8.2. Diversity in protocol design

Protocol design can include decisions that limit - or, encourage the limitation - or service diversity. This has market-facing risks but also has security implications. Where a protocol is too complex to upgrade or implement, only those with the necessary resources are likely to benefit from the deployment. Where larger stakeholders dominate deployment of a particular protocol - because of its underlying design - the result can be a limited pool of deployments with potentially smaller numbers of points of failure.

When protocol choices lead to lack of service diversity, this also increases the risk of shared vulnerabilities among a small set of providers of service. In this case, the attractiveness of the target becomes higher as the number of service implementation becomes lower.

Protocol design should encourage diversity of service provision.

8.3. Protocol design and failure of intermediaries

The rise of intermediaries has been an important feature of the evolution of the Internet. In a prior era, it would have been natural and sufficient to concentrate on deploying security at endpoints. The end-to-end principle seems to encourage thinking of building intelligence - and thus, protection - at the edges of the network.

From previous sections, it is apparent that a strict view of the end-to-end model no longer applies. Even the model for an endpoint, as provided in section 2, no longer can support the view of endpoint protection being for the device or UE. The result is the security of the endpoint is, in part, dependent on the security of the middleboxes and intermediary hosts that provide services.

From the point of protocol design, planning for this includes planning for when those intermediaries are compromised, unavailable or stop delivering expected services correctly. Protocols that depend upon intermediaries should clearly adapt to failure modes of the intermediaries and give choices to endpoint on what actions to take in the event of third-party failure.

8.4. Protocol evolution

A core message in this draft is that: as the Internet evolves, its endpoints do as well, and with them the threats they face. Even ancient protocols are rarely static - especially if they have any deployment base.

As protocols evolve their designers have to achieve a balance between interoperability, strictness, agility and extensibility. Considering this deployment, and altered use case, during design has the property of making protocol evolution more secure.

Care must be taken, especially in complex protocols where many new use cases are being account for, in development of the evolution. It may make the underlying protocol so complex that few are able to implement it securely. Thus, while the protocol itself may be secure, it's complexity makes for error in implementation which leads to operational security problems.

9. Security Considerations

This document is all about security considerations of operational security for endpoints. In particular it provides a model for

endpoint security and then discusses why traditional approaches to endpoint security are no longer sufficient.

10. IANA Considerations

This memo contains no instructions or requests for IANA. The authors continue to appreciate the efforts of IANA staff in support of the IETF.

11. Acknowledgements

The original idea for this draft came from another, now expired draft [33]. The authors of that draft intended a comprehensive discussion of endpoint security and a clear description of how the evolution of the Internet made endpoint security - on its own - insufficient.

The author thanks those previous contributors: Arnaud Taddei, Bret Jordan, Candid Wueest, Chris Larsen, Andre Engel, Kevin Roundy, Yugiong Sun, and David Wells.

The author also extends his appreciation to the discussions in the IAB Activity called model-t where the future of the Internet's threat landscape has also been discussed.

This document was prepared using 2-Word-v2.0.template.dot.

12. References

12.1. Informative References

[1] Thaler, D. and Tschofenig, H., Pei, M., Tsukamoto, A., " Trust Execution Environment Protocol ", draft-ietf-teep-protocol-02 (work in progress), April 2020.

[2] McFadden, M., "Evolution of Endpoint Security - A Taxonomy for Endpoints," draft-mcfadden-opsec-endp-taxonomy-00 (work in progress), February 2021.

[3] "MITRE CAPEC", <https://capec.mitre.org/data/definitions/3000.html>, n.d.

[4] "MITRE ATT&CK", <https://attack.mitre.org>, n.d.

[5] Crotty, J., "New Gartner Report Redefines Endpoints Protection for 2018", January 2018, <https://www.crowdstrike.com/blog/new-gartner-report-redefines-endpoint-protection-for-2018/>.

[6] Redscan, ., "EPP and EDR - What's the difference?", June 2018, <https:/
/www.redscan.com/news/epp-vs-edr-whats-the-difference/>.

[7] Hunt, J., "Advantages and Disadvantages of Three Top Endpoint Security Vendors", n.d., <https://www.adapture.com/blog/evaluating-leading-endpo
int-security-vendors/>.

[8] "IT Pro's Guide to Endpoint Protection", n.d., <https://www.barkly.com/
it-pros-guide-to-endpoint-protection>.

[9] Ericsson, ., "Internet of Things forecast", n.d., <https://www.ericsson
.com/en/mobility-report/internet-of-things-forecast>.

[10] Wueest, C., "How my TV got infected with ransomware and what you can le
arn about it", November 2015, <https://www.symantec.com/connect/blogs/h
ow-my-tv-got-infected-ransomware-and-what-you-can-learn-it>.

[11] Dickson, B., "Millions of smart TVs are vulnerable to hackers", Februar
y 2014, <https://www.dailydot.com/debug/protect-smart-tv/>.

[12] Symantec, ., "W32.Flamer Microsoft Windows Update Man-in-the-Middle", J
une 2012, <https://www.symantec.com/connect/blogs/w32flamer-microsoft-w
indows-update-man-middle>.

[13] Rogers, D., "Handling vulnerabilities as an IoT vendor", December 2018,
<https://www.iotsecurityfoundation.org/less-than-10-of-consumer-iot-com
panies-follow-vulnerability-disclosure-guidelines/>.

[14] Symantec, ., "Mirai, what you need to know about the botnet behind rece
nt major DDoS attacks", October 2016, <https://www.symantec.com/connect
/blogs/mirai-what-you-need-know-about-botnet-behind-recent-major-ddos-a
ttacks>.

[15] Krebs on security, ., "19 Mirai Botnet Authors Avoid Jail Time", Septem

ber 2018, <https://krebsonsecurity.com/tag/mirai-botnet/>.

[16] ESET, ., "LoJax First UEFI rootkit found in the wild, courtesy of the S
ednit group", September 2018, <https://www.welivesecurity.com/2018/09/2
7/lojax-first-uefi-rootkit-found-wild-courtesy-sednit-group/>.

[17] Claburn, T., "Intel SGX safe room easily trashed by white-hat hacking m
arauders Enclave malware demoed", February 2019, <https://www.theregist
er.co.uk/2019/02/12/intel_sgx_hacked/>.

[18] Cimpanu, C., "Researchers hide malware in Intel SGX enclaves", February
2019, <https://www.zdnet.com/article/researchers-hide-malware-in-intel-
sgx-enclaves/>.

[19] Khandelwal, S., "Explained - How Intel AMT Vulnerability Allows to Hack Computers Remotely", May 2017, <https://thehackernews.com/2017/05/intel-amt-vulnerability.html>.

[20] Symantec, ., "Web Attack Intel AMT Privilege Escalation CVE-2017-5689", 2017, <https://www.symantec.com/security_response/attacksignatures/detail.jsp?asid=29888>

[21] Smith, K., "9 signs your endpoint security isn't working", May 2017, <https://securelement.com/9-signs-your-endpoint-security-isnt-working/>.

[22] Cobb, M., "SQL injection detection tools and prevention strategies", November 2009, <https://www.computerweekly.com/tip/SQL-injection-detection-tools-and-prevention-strategies>.

[23] Wueest, C., "Living off the land and fileless attack techniques", July 2017, <https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-living-off-the-land-and-fileless-attack-techniques-en.pd>.

[24] Caballero, J., "Mind Your Own Business A Longitudinal Study of Threats and Vulnerabilities in Enterprises", February 2019, <https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_03B-1-2_Kotzias_paper.pdf>

[25] McHugh, J., "Windows of Vulnerability A Case Study Analysis", 2000, <http://www.cs.colostate.edu/~cs635/Windows_of_Vulnerability.pdf>.

[26] Plattner, B., "Large-Scale Vulnerability Analysis", September 2006, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.3056&rep=rep1&type=pdf>.

[27] Marinho, R., "Exploring a P2P transient botnet - From Discovery to Enumeration", July 2017, <https://morphuslabs.com/exploring-a-p2p-

transient
-botnet-from-discovery-to-enumeration-e72870354950>.

[28] Seals, T., "Pulse-Wave DDoS Attacks Mark a New Tactics in Q2", October 2017, <https://www.infosecurity-magazine.com/news/pulsewave-ddos-attack
s-mark-q2/>.

[29] UK, GOV., "Secure by Design", February 2019, <https://www.gov.uk/govern
ment/collections/secure-by-design>.

[30] Cullen, T., "Limits of endpoint only", July 2017, <https://www.adapture
.com/blog/evaluating-leading-endpoint-security-vendors/>

[31] Dix, J., "Layered Security Defenses What layer is most critical network or endpoint", July 2011, <https://www.networkworld.com/article/2220204/tech-debates/layered-security-defenses--what-layer-is-most-critical--network-or-endpoint-.html>.

[32] Hstoday, ., "Layered Approach Critical to Effective Endpoint Protection", October 2016, <https://www.hstoday.us/channels/federal-state-local/layered-approach-critical-to-effective-endpoint-protection/>.

[33] Taddei, A., et.al., [I-D:draft-taddei-smart-cless-introduction], "Capabilities and Limitations of an Endpoint-only Security Solution, July 2020, (expired).

Appendix A. Operational Experience and Endpoint Security

This appendix attempts to document operational experience related to endpoint security. The original text in this Appendix comes from research and experienced from a single security solution provider. What follows is reporting from that organization.

There are two appaorches to reporting on security incidents that are based on operational experience. They are:

o the method described in {{MONEYBALL}}

o the anonymised production data of Symantec MSS production for the past 3 months

The core idea is to consider, based on all the imperfections we started to list above including the 'grey areas', that cybersecurity analysts are often presented with suspicious machine activity that does not conclusively indicate a compromise, resulting in undetected incidents or costly investigations into the most appropriate remedi

As Managed Security Services Providers (MSSP's) are confronted with these data quality issues, but also possess a wealth of cross-product security data that enables innovative solutions, we decided to use the Symantec MSS service for the past 3 months. The Symantec MSS service monitors over 100 security products from a wide variety of security vendors for hundreds of enterprise class customers from all verticals.

We selected the subset of customers using the service that deploy both network and endpoint security products to determine which types of security incidents were most likely to be detected by endpoint products vs. network products. In doing so, we were particularly interested in identifying which categories of incidents are detected by endpoint products and not network products, and vice versa. Thus, we examined prevalent categories of incidents for which the only actionable security alerts were predominantly produced by one type of security product and not the other. To do so, we extracted all security incidents detected by Symantec MSS on behalf of hundreds of customers that deploy both network and endpoint security products, over a three-month period from December 2018 through the end of February 2019. We acknowledge that some attacks might have been blocked by the first product and therefore have never been seen by the next security solution, which influences the final numbers.

With this in mind, we could identify incidents based on:

| Severity | 4 - Emergency, 3 - Critical, 2 - Warning, 1 - Informational |

| Incident Category | Malicious Code, Deception Activity, Improper Usage, Investigation, etc. |

| Incident Type | Trojan Horse Infection, Suspicious DGA Activity, Suspicious Traffic, Suspicious URL Activity, Backdoor infection, etc. |

| # network incidents | Amount of network only security incidents |

| # all incidents | What is the total amount of incidents on all security solutions |

| Percentage | Percentage of network security only incidents |

We ended up with

- Hundreds of thousands of security incidents

- which we could categorize in 275 incident types by category and severity (triplets Severity-Category-Type)

- out of which we searched how many incidents of each type were detected by a network security product and missed by deployed endpoint security products at least 75% of the time or vice versa

A.1. Endpoint only incidents

The categories of incidents that are detected primarily by endpoint security products are fairly intuitive. They consist primarily of detections of file-based threats and detection of malicious behaviors through monitoring of system and network behavior at the process level. The most prevalent of these behavioral detections include detections of suspicious URLs based on heuristics and blacklists of IP addresses or domain names. Since most of these alerts are not corroborated by network products, it seems probable that the blacklists associated with network products tend to be more

focused on attacks while host-based intrusion prevention system alerts focus more on malware command and control traffic. Most other behavioral detections at the endpoint provide alerts based on system behavior that is deemed dangerous and symptomatic of malicious intent by a malicious or infected process. The highest severity incidents detected on endpoints are instances of post-compromise outbound network behavior that are symptomatic of command and control communications traffic, but these did not show up as being primarily detected by endpoint products as they are frequently corroborated by network-based alerts.

A.2. Security incidents detected primarily by network security products

Perhaps less intuitive are the results of examining categories of security incidents that are detected primarily by network security products and only rarely corroborated by endpoint security products. Below we provide details regarding incident categories for which a network security product produced a detection and for which there were no actionable endpoint alerts for at least 75% of the incidents in the category.

In our study we found 32 incident type, category, and severity triplets of this type. The following categories critical incident types were reported by MSS customers, and we discuss each in turn in decreasing order of prevalence:

A.2.1. Unauthorized external vulnerability scans

Perhaps unsurprisingly, unauthorized external attempts to scan corporate resources for vulnerabilities and other purposes are detected in large volumes by a broad variety of network-focused security products. 79% of incidents of this type were detected by network security products with critical-severity alerts, these security incident detections are not accompanied by any actionable endpoint alerts, despite the fact that endpoint security products are deployed by these enterprises. This category of threats encompasses a broad variety of attacks, the most prevalent of which are the following: Horizontal scans, SQL injection attacks, password disclosure vulnerabilities, directory traversal attacks, and blacklist hits. Of these categories of detections, horizontal scans stand out as the category of detection that endpoint-security products are least likely to detect on their own.

A.2.2. Unauthorized internal vulnerability scans

Unauthorized internal vulnerability scans, though less frequent, are more alarming, as they are likely to represent possible post-

compromise activity. We note that the Managed Security Service works with its customers to maintain lists of devices that are authorized to perform internal vulnerability scans, and their activity is reported separately at a lower levels of incident severity. 89% of detected unauthorized internal vulnerability scans are detected by network products without any corroborating actionable alerts from endpoint security products. As compared to unauthorized external scan incidents, internal hosts that perform vulnerability scans are far more active and the fraction of alerts that detect horizontal scans is higher, representing half of the total alerts generated. Alerts focused on Network-Behavior Anomaly Detection also appear for internal hosts.

A.2.3. Malware downloads resulting in exposed endpoints

This category of threats is generally detected by network security appliances. Despite these enterprises being purchasers of endpoint security products, 76% of the incidents detected by the network security products do not show a corresponding alert by an endpoint security product. A broad variety of network appliances contributed to the detection of a diverse collection of malware samples.

A.2.4. Exploit kit infections

This category of infections represents instances in which the customer's machines are exposed to exploit kits. These threats were detected by network appliances that extract suspicious URLs from network traffic taps and use a combination of sandbox technology and blacklists to identify websites that deploy a variety of exploit kits that were not being caught by endpoint security products. In this three month time period, the most prevalent categories of exploit kits detected involved redirections to the Magnitude exploit kit and exploit kits associated with phishing scams and attempts to expose users to fake Anti-Virus warnings and tools. A breakdown of the results is included below:

| Severity | 3 - Critical |

| Incident Category | Malicious Code |

| Incident Type | Exploit Kit Infection |

| # network incidents | 26 |

| # all incidents | 26 |

| Percentage | 100% |

The network security product that detected these incidents produced the following alerts:

- Advanced Malware Payloads
- Exploit.Kit.FakeAV
- Exploit.Kit.Magnitude
- Exploit.Kit.MagnitudeRedirect
- Exploit.Kit.PhishScams
- HTMLMagnitudeLandingPage

A.2.5. Attacks against servers

In addition to detecting the aforementioned critical security incident categories, network security devices frequently detect a broad variety of attacks against servers that usually lack corroboration at the endpoint. Most server attacks are not matched by endpoint protection alerts: 62% are unmatched for critical incidents, and 88% are unmatched as lower severity incidents. This category of incidents is the most prevalent category of incidents detected primarily by network products, but they are usually rated lower in severity than the aforementioned classes of alerts as they are very commonplace. Even when these alerts are corroborated by endpoint protection alerts, the endpoint alerts are often low in severity, as in the case of file-based threats that appear to have been blocked or successfully cleaned up by an Anti-Virus solution. The challenge in taking action against server attacks is that it can be difficult to assess which of these attacks were successful in causing actual damage, and for this reason, for the fraction of server attacks that demonstrate corroborating endpoint security alerts, even if of low severity, should be examined. It is interesting to note the cooperative role played by both network and endpoint security devices in these instances.

Authors' Addresses

Mark McFadden
Internet policy advisors ltd uk
6 Bridge Street
Chepstow, Wales NP16 5EY
United Kingdom

Phone: +44 2921 25 3649
Email: mark@internetpolicyadvisors.com