PERCEPTION POINT

Malware is a red herring.

The real enemy is its source.

WHO AM I?

SHLOMI LEVIN
Co-founder & CTO

- Formerly, Senior Technology Leader at Cyvera (acquired by Palo Alto Networks) and at Trusteer (acquired by IBM)
- Officer & Research Squad Leader in classified elite cyber unit in the Israeli Intelligence Corps

AGENDA

1
Exploits & Exploitation Techniques

2
Why It's Important

3
The Evolution of Exploitation Techniques

4
Evasion Techniques

5
Our Approach & Offering

1

Exploits & Exploitation Techniques

FIRST THINGS FIRST:
DEFINITIONS

Exploit

- Code that leverages a software bug (**vulnerability**) to infect a system
- In simple words: the trigger that enables the attacker to deliver the malware

Exploitation Technique

- A limited set of techniques used to conduct an exploit
- Typically developed in academia
- Exploitation techniques are the technical actors behind advanced threats

FIRST THINGS FIRST: DEFINITIONS (CONT'D)

Zero-day

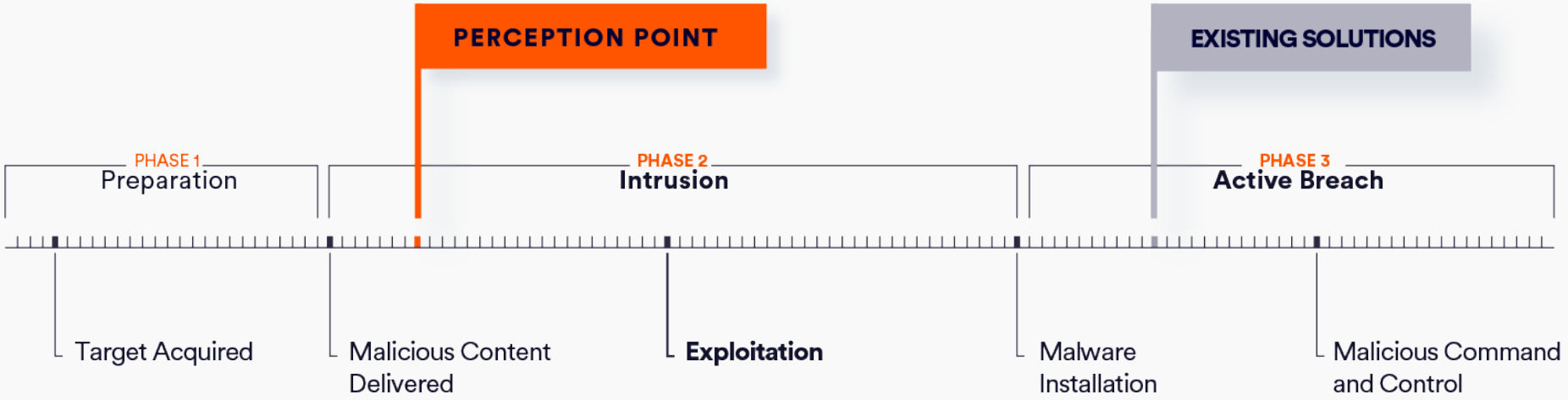
- A cyber attack leveraging software bugs that are **completely unknown** and have no patch

N-day

- A cyber attack leveraging software bugs that are known (Usually published by cybersecurity companies)
- Altered signatures prevent detection

EXPLOITS & EXPLOITATION TECHNIQUES

EXPLOITS ARE THE DELIVERY VEHICLE FOR MALWARE



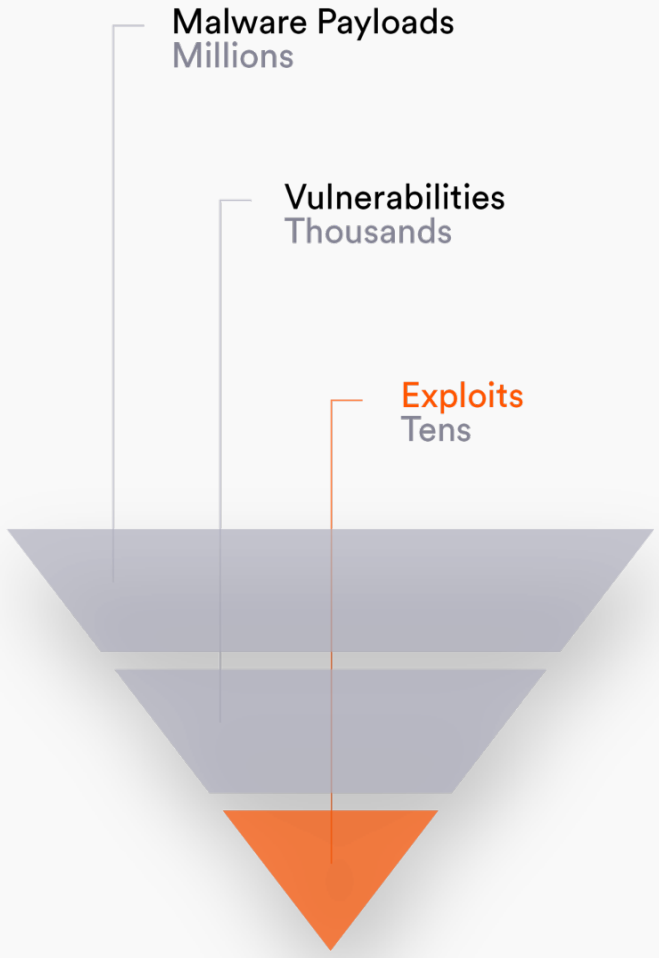
- Exploitation is a **deterministic** act that happens earlier in the kill chain, **pre-malware release**.

PERCEPTION POINT

EXPLOITS & EXPLOITATION TECHNIQUES

IT'S A NUMBER GAME. IS IT?

- In advanced attacks exploits are the real enemy.
- While there were 670 MM new malware variants in 2017 (+88% YoY) [1], there are only a limited number of exploits.

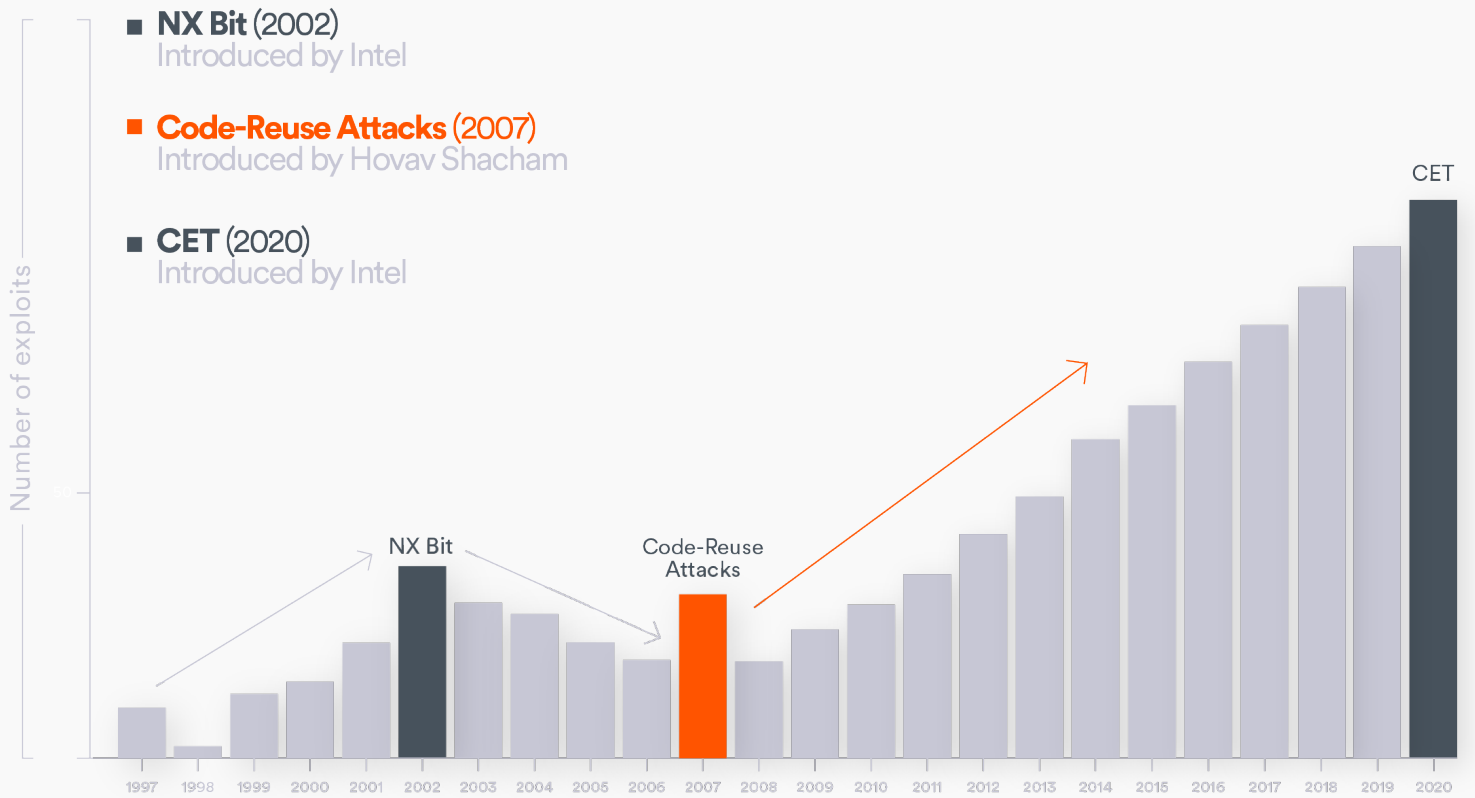


Note:
1. Symantec ISTR March 2018

PERCEPTION POINT

EXPLOITS & EXPLOITATION TECHNIQUES

KEY MILESTONES IN THE EXPLOIT'S EVOLUTION



PERCEPTION POINT

2

Why It's Important

WHY IT'S IMPORTANT

THE THREAT LANDSCAPE: ADVANCED THREATS OVERVIEW

Memory Corruption

- Transferred wither by files or links
- Attack techniques include: Heap Spray, ROP, COP and more

Logical Bugs / Droppers

- Exploiting **logical bugs** in a software and/or **features** for malicious purposes
- E.g. mouse-hovering, DEE

Payload-less Attacks

- Attacks that don't include any file or links
- Includes BEC & ATO attacks

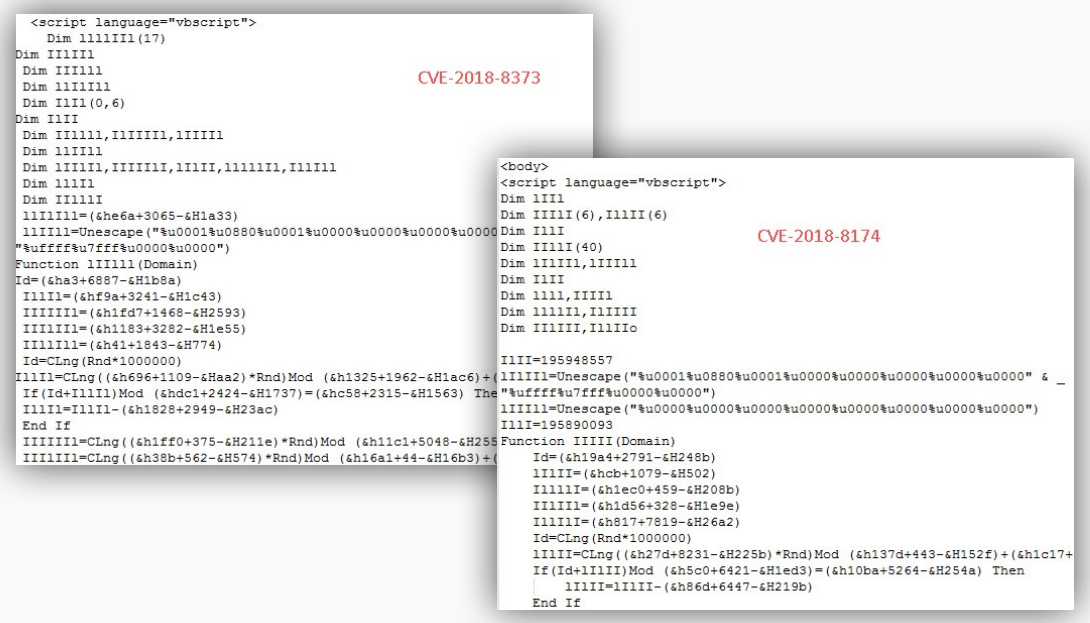
PERCEPTION POINT

WHY IT'S IMPORTANT

EXAMPLE #1:
INTERNET EXPLORER VULNERABILITY

Attack Overview

- **The attacker:** Darkhotel, a North Korea-based APT group
- **The vulnerability:** CVE-2018-8373, a Zero-day in **Internet Explorer 9, 10, and 11**
- The flaw could be exploited by remote attackers to take control of the systems by tricking victims into viewing a specially crafted website through Internet Explorer.



Analysis of the exploit code revealed it shared the obfuscation technique implemented for another flaw (CVE-2018-8174)

Sources: Trend Micro & Security Affairs

PERCEPTION POINT

WHY IT'S IMPORTANT

EXAMPLE #2:
3 MS OFFICE'S EPS ZERO-DAYS (CVE-0261/0262/0263)

Attack Overview

- **The attackers:**
  - **Turla** – a Russian cyber espionage APT group
  - **APT28** – a Russian cyber espionage APT group
  - **A new unknown financially motivated actor**

- **The targets:** European diplomatic and military entities and regional and global banks with offices in the Middle East.

- The exploits leveraged 3 vulnerabilities in Microsoft Office Encapsulated PostScript (EPS)

Source: FireEye

PERCEPTION POINT

WHY IT'S IMPORTANT

EXAMPLE #3:
SHARED DRIVE ATTACK

Attack Overview

- **The attacker:** MUSTANG PANDA, a China-based hacking group
- **The target:** Mongolia-based victims
- The attack involved the use of shared malware.
- The group used a series of redirections and **file-less**, malicious implementations of legitimate tools to gain access to the targeted systems.

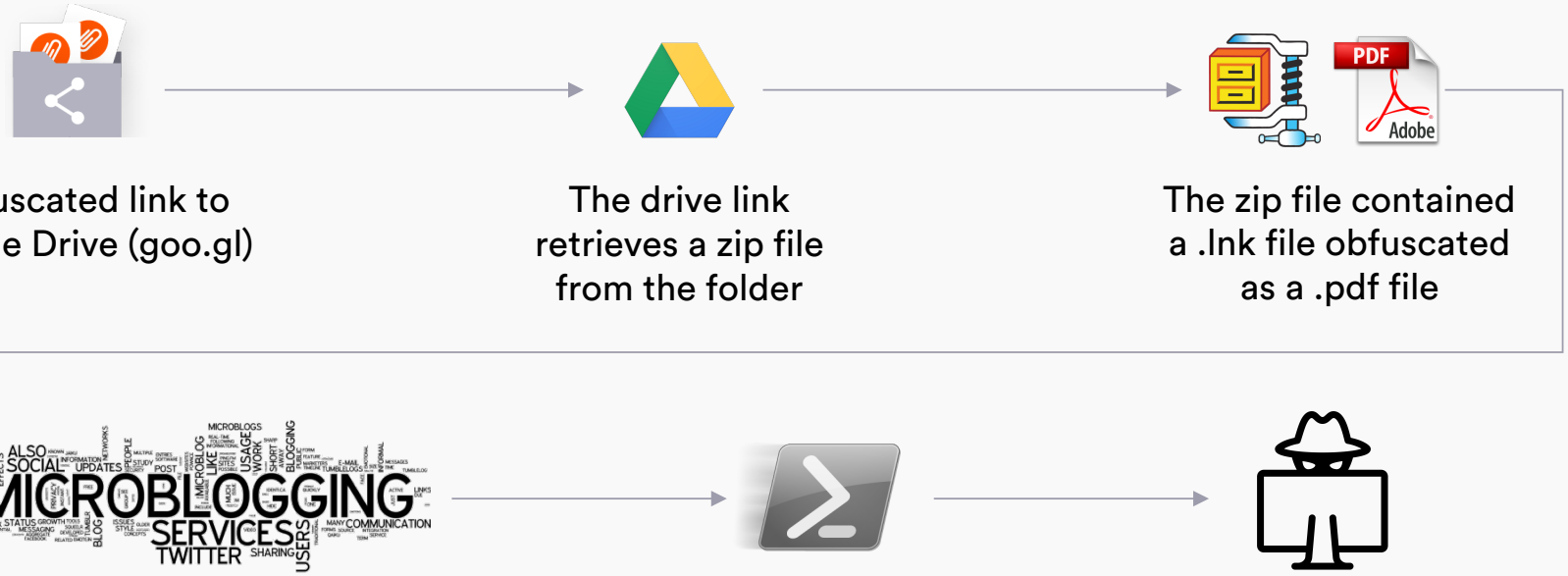
Source: CrowdStrike

PERCEPTION POINT

WHY IT'S IMPORTANT

EXAMPLE #3: SHARED DRIVE ATTACK (CONT'D)

The Attack Chain



Obfuscated link to Google Drive (goo.gl)

The drive link retrieves a zip file from the folder

The zip file contained a .lnk file obfuscated as a .pdf file

The .lnk file redirects the user to a .wsc file hosted on a micro-blogging page controlled by the attacker

The file uses VBScript to retrieve a decoy PDF file and a PowerShell script

The attacker runs the malware and gains control on the target

Source: CrowdStrike

PERCEPTION POINT

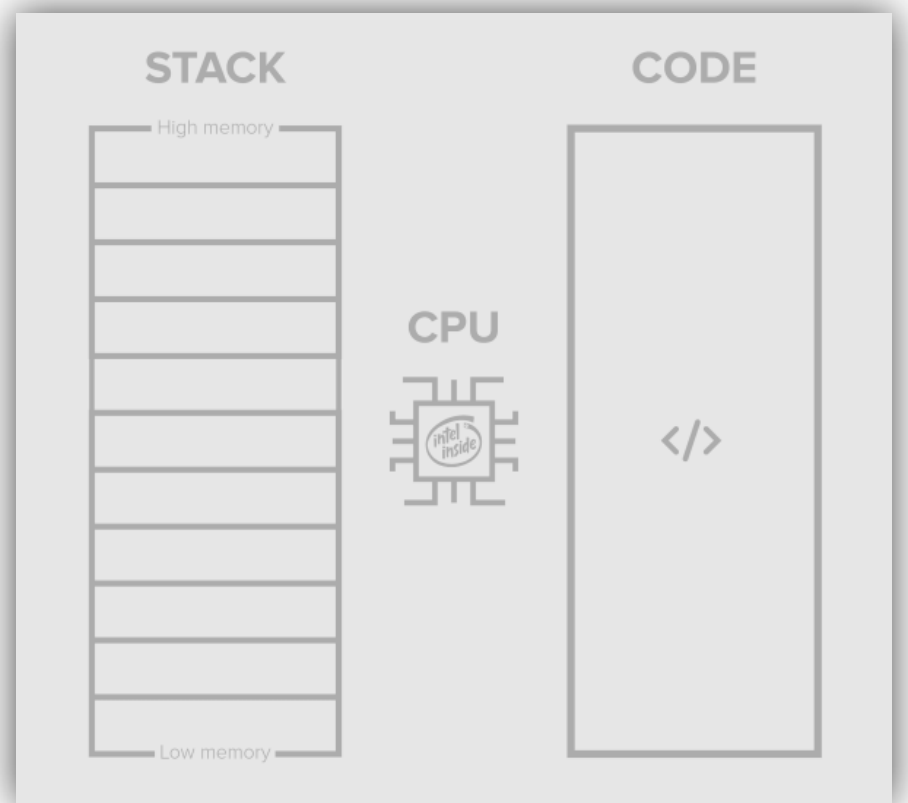
3

The Evolution of Exploitation Techniques

THE EVOLUTION OF EXPLOITATION TECHNIQUES

THE 90'S: CODE INJECTION ATTACKS

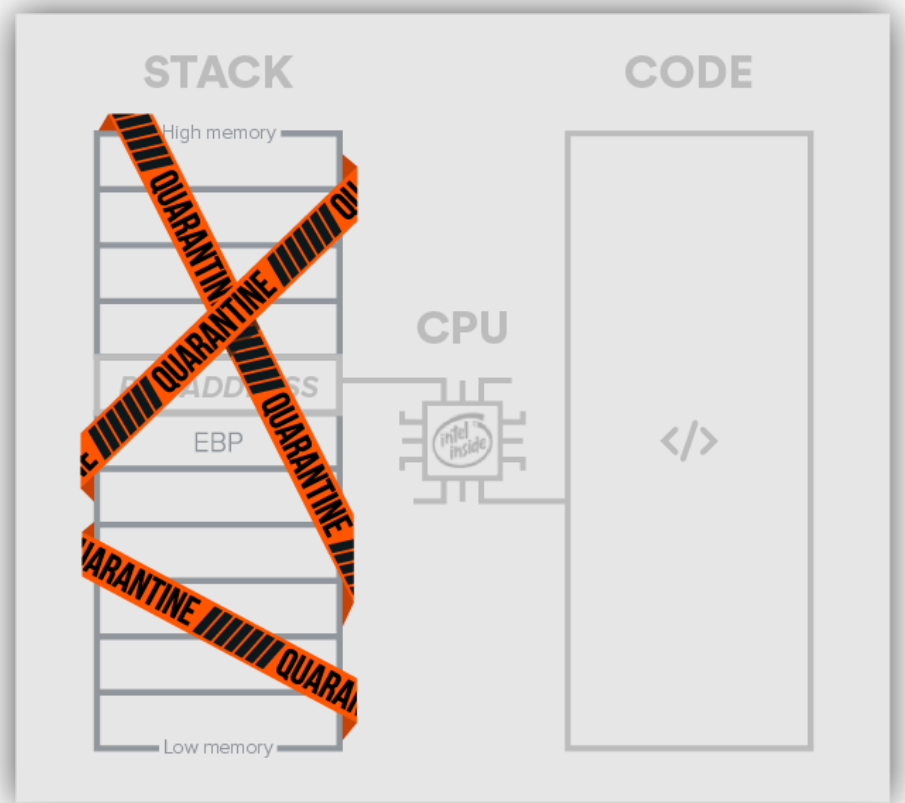
- "RET" opcode is tricked and returns to malicious code.
- Injected to the software by the attacker.
- These techniques were very successful for a time as there were no defensive measure in place.



PERCEPTION POINT

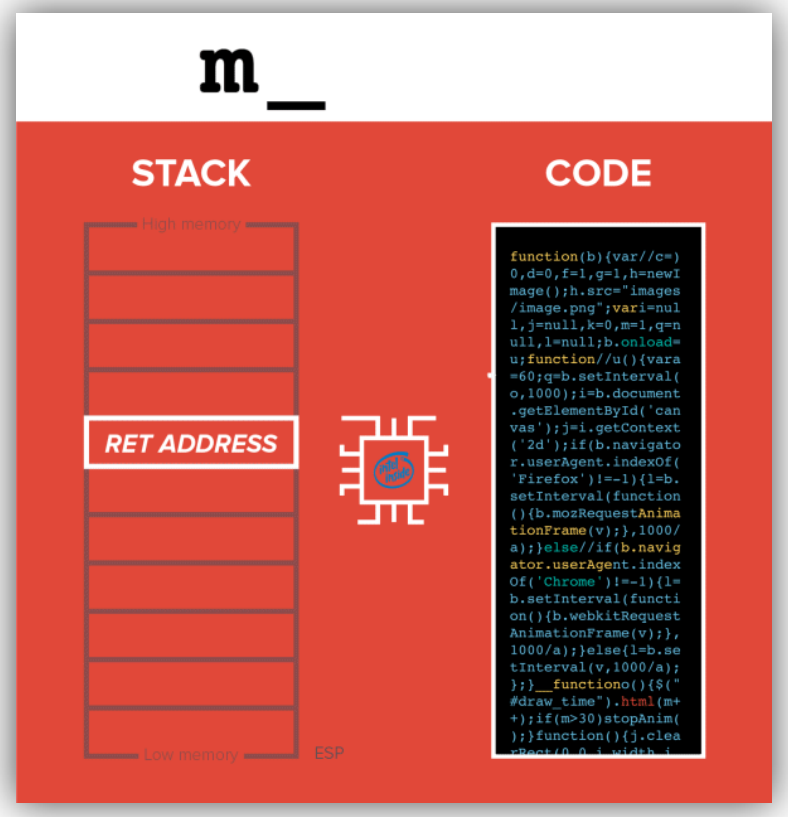
INTEL'S NX (NO-EXECUTE) BIT

- Hardware protection prevents attackers from injecting and executing malicious code from the stack.
- Life got hard for attackers for 6 years as the new Intel CPU's were widespread.



THE 2000'S: CODE REUSE ATTACKS

- If injected code can't be executed use **EXISTING** code instead!
- ROP was introduced in 2007 to bypass NX Bit. With ROP, an attacker chains small pieces from the normal code (gadgets) – to construct a new malicious code.
- For 10 years there's been no protection against Code-Reuse Attacks such as ROP, resulting in an exponential increase in exploits.



PERCEPTION POINT

THE EVOLUTION OF EXPLOITATION TECHNIQUES

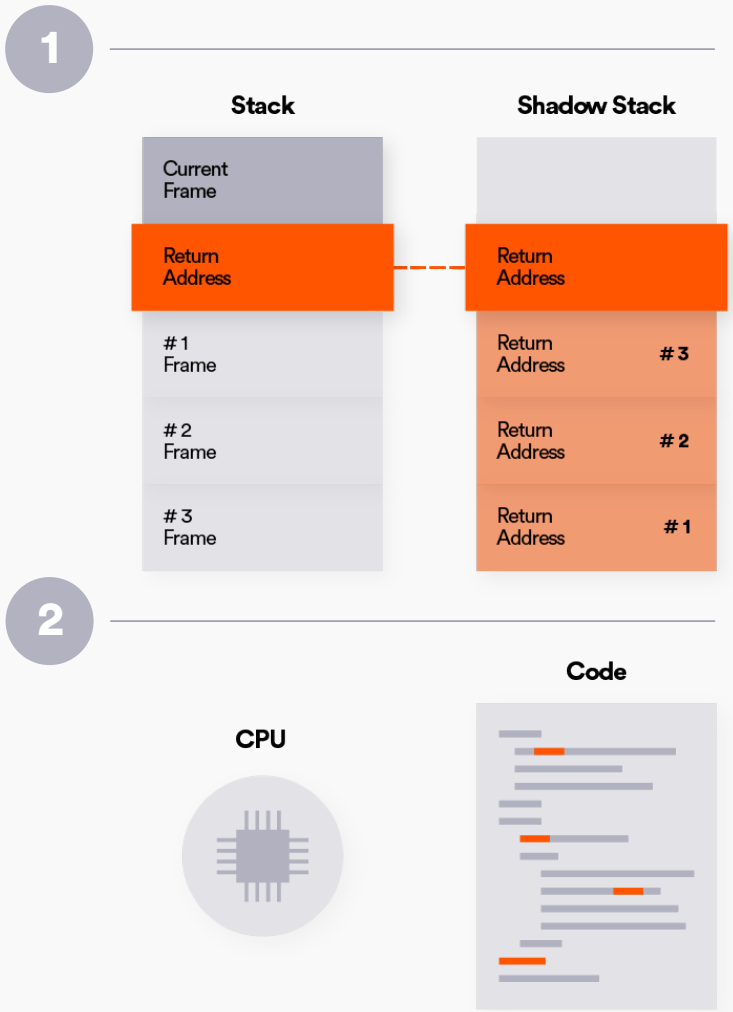
INTEL CONTROL-FLOW ENFORCEMENT TECHNOLOGY (CET)

- Hardware protection that provides the following capabilities to defend against code reuse attacks:

1 **Shadow Stack**
Return address protection to defend against Return Oriented Programming.

2 **Shadow Stack**
Return address protection to defend against Return Oriented Programming.

The (1st) problem: to be released only in 2020 (TBD)

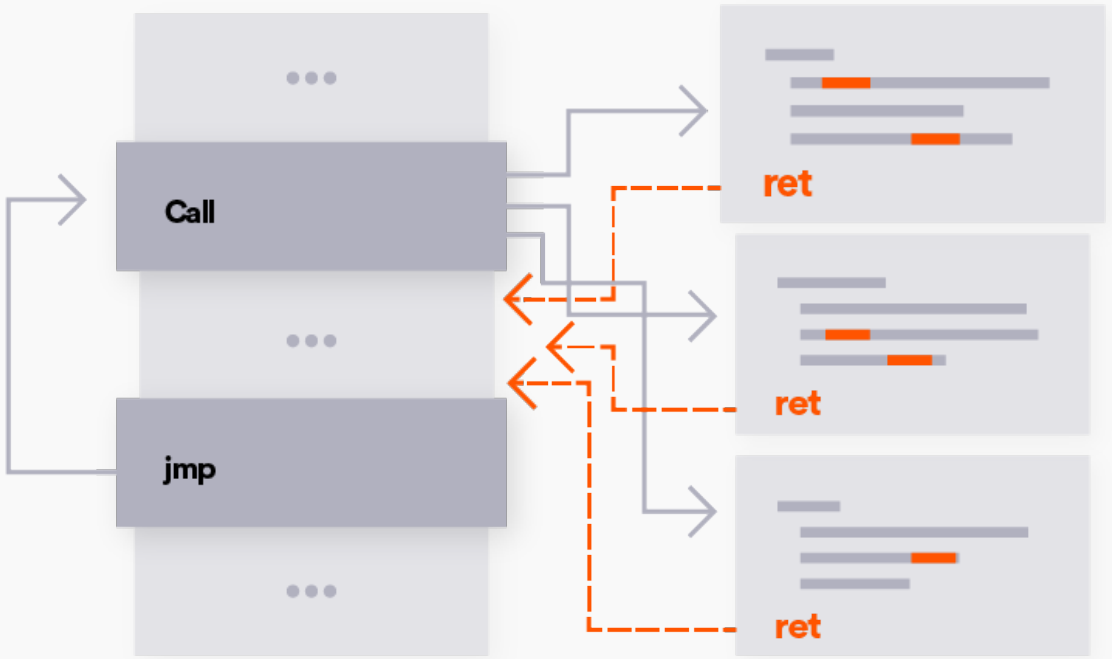


PERCEPTION POINT

THE EVOLUTION OF EXPLOITATION TECHNIQUES

AND MORE BAD NEWS... ADVANCED CODE RE-USE ATTACKS

- If small code fragments can't be executed use **LEGITIMATE** code such as functions and virtual functions.
- Techniques such as **LOP**, **DOP** and **COOP** essentially setup a loop gadget to invoke a series of legitimate functions to carry out malicious computations.



PERCEPTION POINT

4

Evasion Techniques

EVASION TECHNIQUES: HOW HACKERS BYPASS SANDBOXES

1

Embedding the Payloads

- Deep, sophisticated packaging
- Clicked when triggered

2

Detecting the Existence of a Sandbox

- The code runs differently in the virtualized environment

3

Exploiting the Sandbox's vulnerabilities

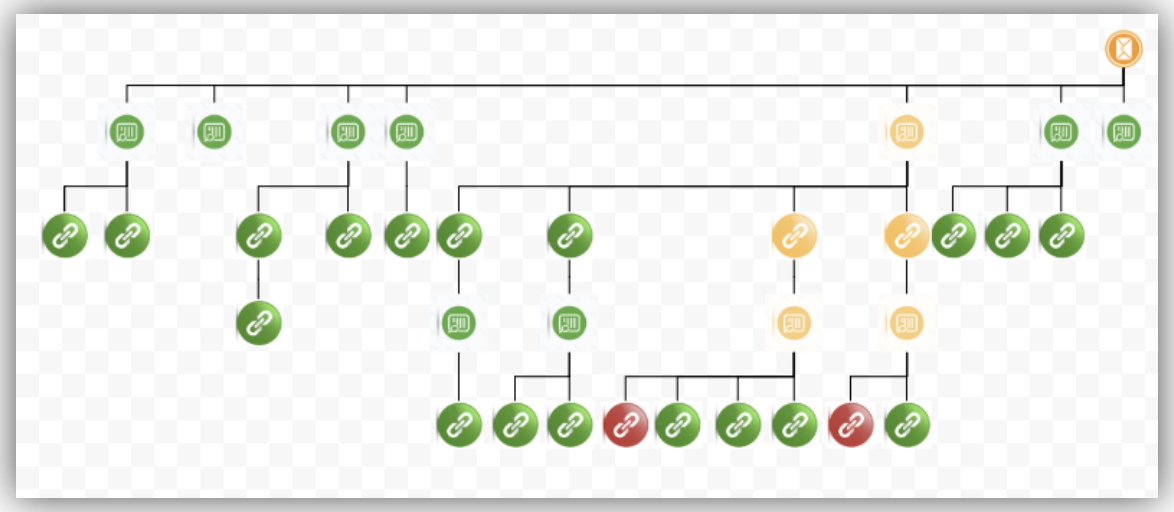
- Cutting the hooks
- Scale and sizes of files

EVASION TECHNIQUES (CONT'D)

1 Packaging

- Attackers simply conceal the malicious payload by deeply embed them within other files or links, taking advantage with the scale problem of sandboxes.

- This evasion is pretty easy and does not require any advanced hacking capabilities.



Based on attack caught in Perception Point's system

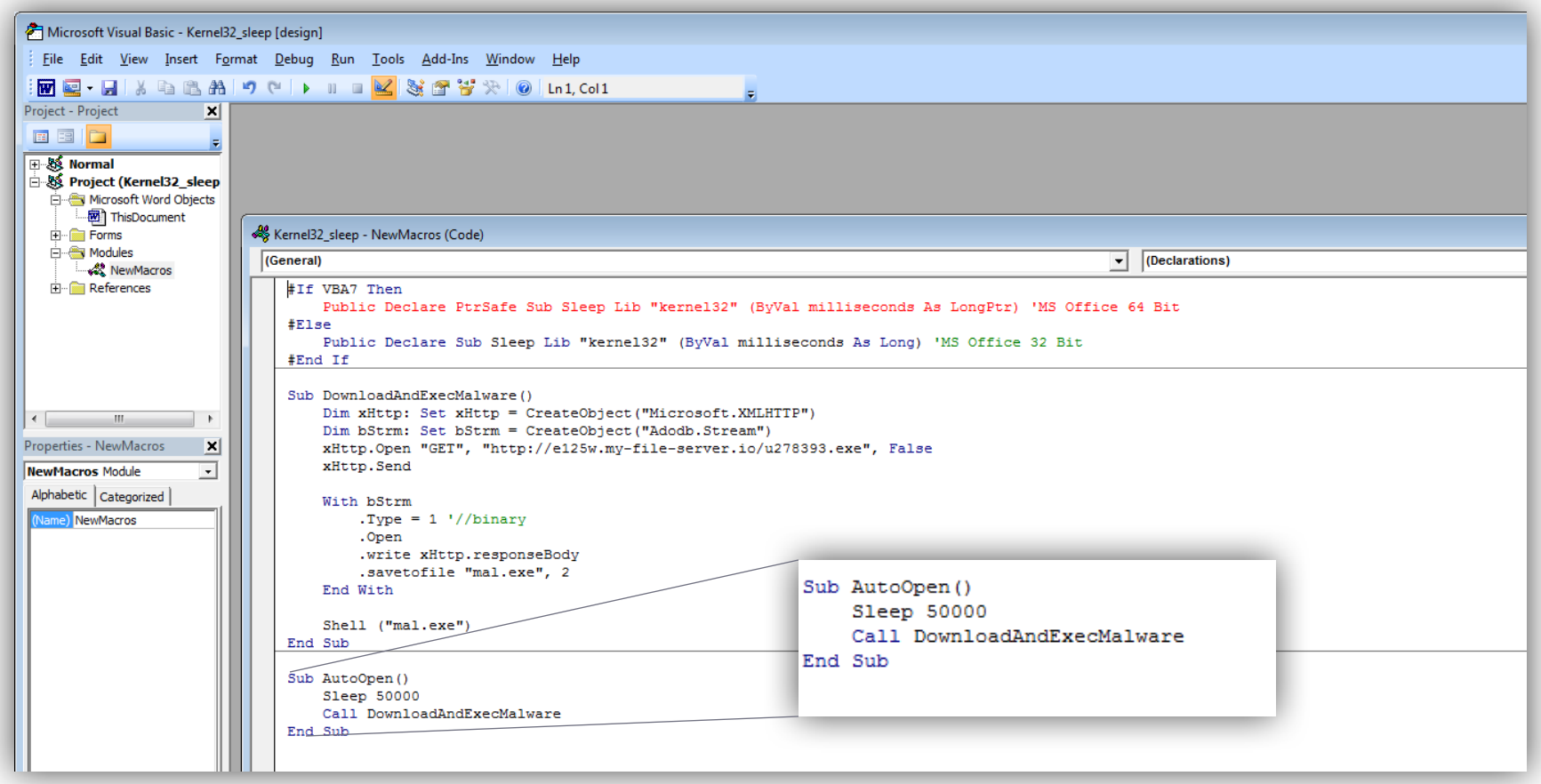
EVASION TECHNIQUES (CONT'D)

2 Sleepers

- Since many sandboxes have limit time in which they test a file/link, many attackers insert a sleeper of several minutes to bypass the defense layers.

- Again, this technique requires minimal hacking capabilities.

EVASION TECHNIQUES (CONT'D)



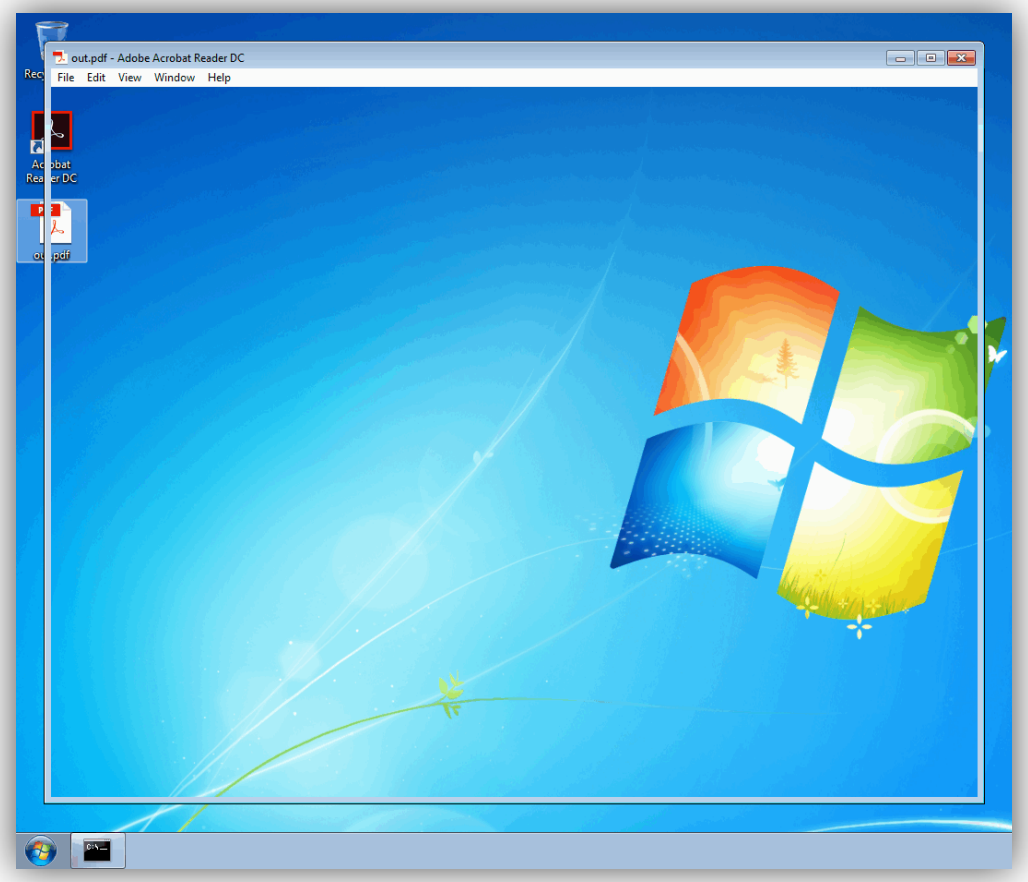
PERCEPTION POINT

EVASION TECHNIQUES (CONT'D)

3 Detecting The "Artificiality" of The Environment

- Detecting that the environment is not real in the sense **a human does not use it**.

- Examples include: checking screen resolution, drivers, memory size, system uptime, cookies, desktop icons, languages, time zones, and more.



Sources: VMRay, Perception Point

EVASION TECHNIQUES (CONT'D)

```
1   function CVE_2018_4990_trigger() {
2       var f = this.getField("Button1");
3       if(f){
4           f.display = display.visible;
5       }
6   }
7
8   app.monitors.toSource();
9
10  var number_of_connected_monitors = app.monitors.length;
11
12  if (number_of_connected_monitors >= 2) {
13      CVE_2018_4990_trigger();
14  }
```

EVASION TECHNIQUES (CONT'D)

4 Detecting Virtualization / Hypervisor

- Detecting technical artifacts that exist due to the lack of full hardware support for virtualization.
- Examples include: detecting artifacts of popular VM hypervisors, or detecting generic hypervisor artifact.
- This type of attack usually takes place only after **the exploit is being run,** i.e. as part of the **malware execution**.

Sources: VMRay, Perception Point

EVASION TECHNIQUES (CONT'D)

5 Detecting Sandbox Artifacts

- Detecting the **sandbox itself** (vs. the hypervisor).
- In this approach, the hacker can utilize the fact that **a sandbox has hooks** – a layer capturing communication between processes, drivers and the OS.
- Again, this type of attack usually takes place only after **the exploit is being run,** i.e. as part of the **malware execution**.

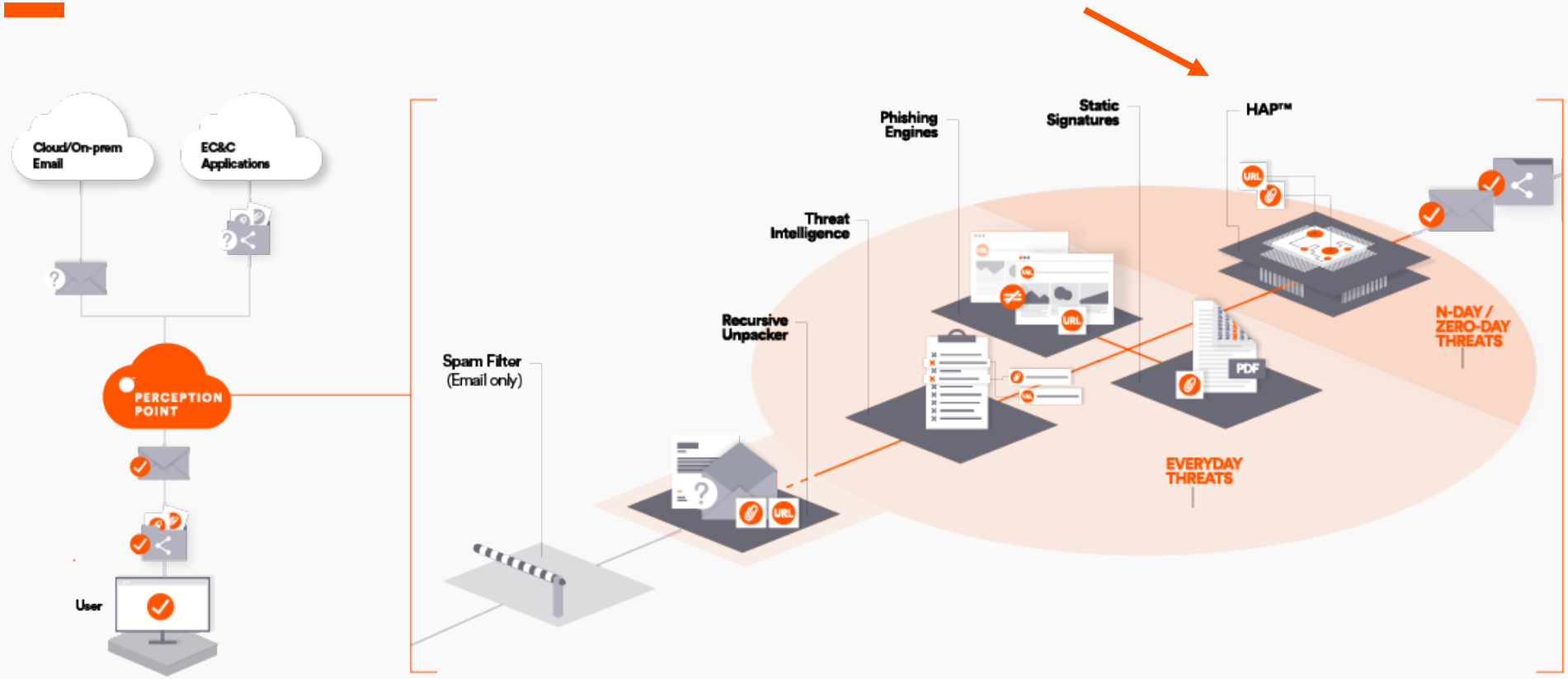
Sources: VMRay, Perception Point

5

Our Approach & Offering

OUR APPROACH & OFFERING

ADVANCED EMAIL & SHARED DRIVES PROTECTION



PERCEPTION POINT

OUR APPROACH & OFFERING

THE HAP™: FIRST-EVER HARDWARE ASSISTED PLATFORM BLOCKS ZERO-DAY AND N-DAY EXPLOITS

Key Goal:

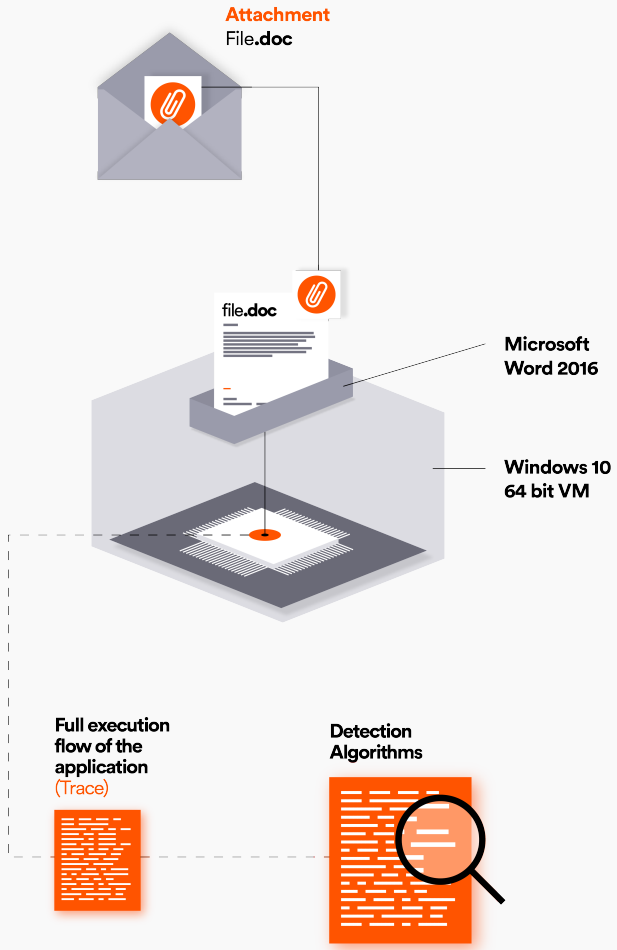
Provide real-time prevention by intercepting malicious documents and URLs that leverage:

- Zero-day vulnerabilities
- N-day vulnerabilities targeting unpatched software updates
- Never-seen-before malicious document with various scripts (e.g. Word macros)

How We Address It:

Software algorithms use **CPU level data** to access the entire execution flow, right from the processor.

Deterministically intercepts exploit techniques pre-malware release.

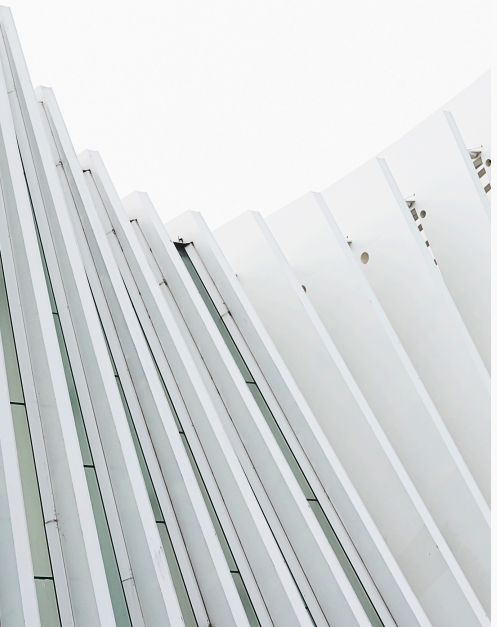


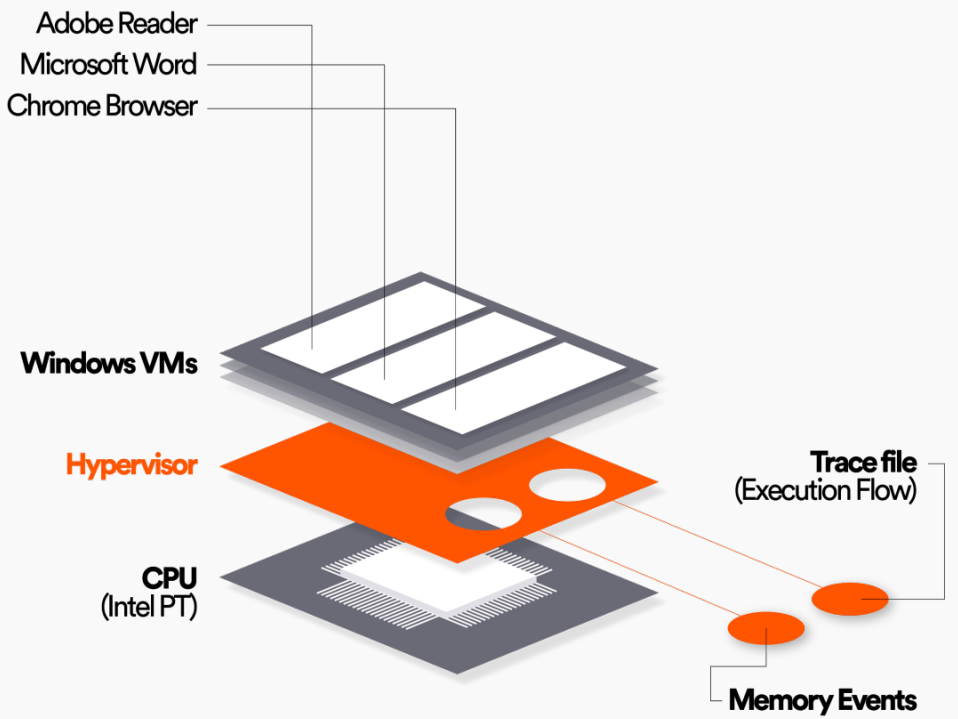
PERCEPTION POINT

OUR APPROACH & OFFERING

THE HAP™: HARDWARE VISIBILITY

- **Leverage Intel PT (Processor Trace), to gain access to the full execution flow of an application**.
- Custom built hypervisor used as a bridge between the hardware and the virtual machines (VMs) that detonated the files/URLs.
- When a file is running inside a VM, its full execution flow is recorded (creating a trace file) together with changes to virtual memory during execution.
- This together with the memory events is then fed to the scanners (detection algorithms) to detect malicious execution flow.





PERCEPTION POINT

OUR APPROACH & OFFERING

THE HAP™: SOFTWARE AGILITY

1

CFG.

- Detects Zero-day & N-day **memory corruption** exploits
- Records the CPU while it processes the input (files and URLs) and identifies exploits by examining the entire execution flow – detecting any deviation from the normal flow of a program in order to deterministically identify malicious activity.

2

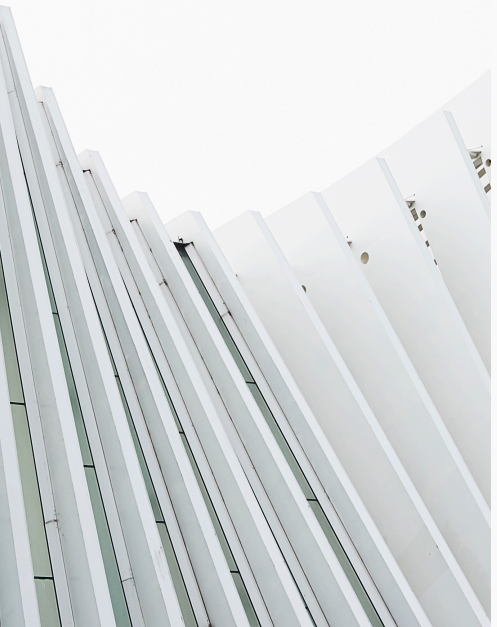
FFG.

- Detects advanced techniques, such as **exploits that are written to bypass common CFI algorithms**.
- Proprietary semantic aware control flow graphs developed for each app identify deviations of the execution flow during runtime.

3

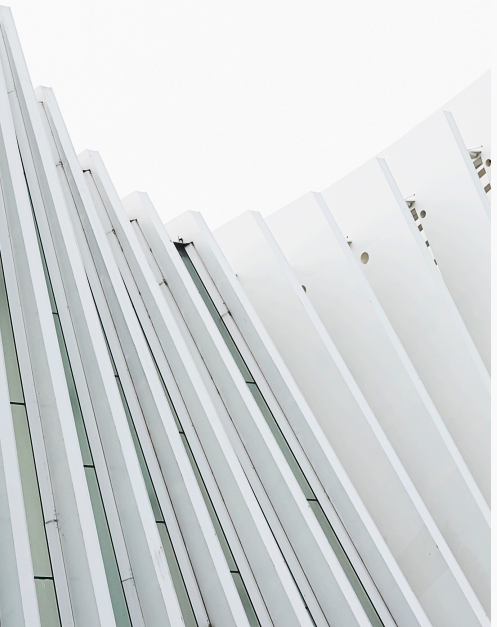
Dropper

- Detects logical **bugs & Droppers** in applications and malicious macros in office documents.
- Employs advanced heuristics-based engine.



PERCEPTION POINT

WHY IT'S IMPORTANT



THE THREAT LANDSCAPE: EXPLOIT TECHNIQUES

Exploit Technique	Year	Individual / Organization
Stack Overwrite Return Address	1996	Elias Levy (also known as Aleph One; a cyber security expert and blogger)
Stack Overwrite Variables	n.a.	n.a.
Stack SEH Overwrite	2003	David Litchfield (NGS Software)
Heap Spray	2004	SkyLined (a well-known blogger)
Stack Pivot	n.a. (1)	n.a.
Return Oriented Programming (**ROP**)	2007	University of California, San Diego
Jump Oriented Programming (**JOP**)	2010	North Carolina State University
Call Oriented Programming (**COP**)	2014	University of California, Berkeley
Counterfeit Object-Oriented Programming (**COOP**)	2015	Ruhr-Universitat Bochum & Technische Universität Darmstadt
Data Oriented Programming (**DOP**)	2016	National University of Singapore

Note:
1. This technique is highly connected to the ROP exploit.

PERCEPTION POINT

OUR APPROACH & OFFERING

EVERYDAY THREATS: OUR COVERAGE

1

Spam Filter

Receives the email & applies reputation and anti-spam filters to quickly flag an email as malicious.

2

Recursive Unpacker

Unpacks the email into smaller units (files and URLs) to identify hidden malicious attacks.
Further extracts embedded URLs and files (recursively) by unpacking files and following URLs.

3

Threat Intelligence

Combines multiple threat intelligence sources with our internally developed engine that scans URLs and file in the wild to warn about potential or current attacks.

4

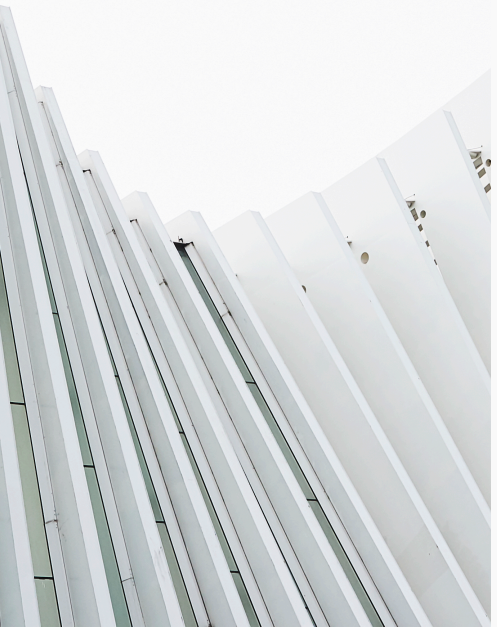
Phishing Engines

Combines best-in-class URL reputation engines and an in-house image analysis engine to identify impersonation techniques and phishing attacks.

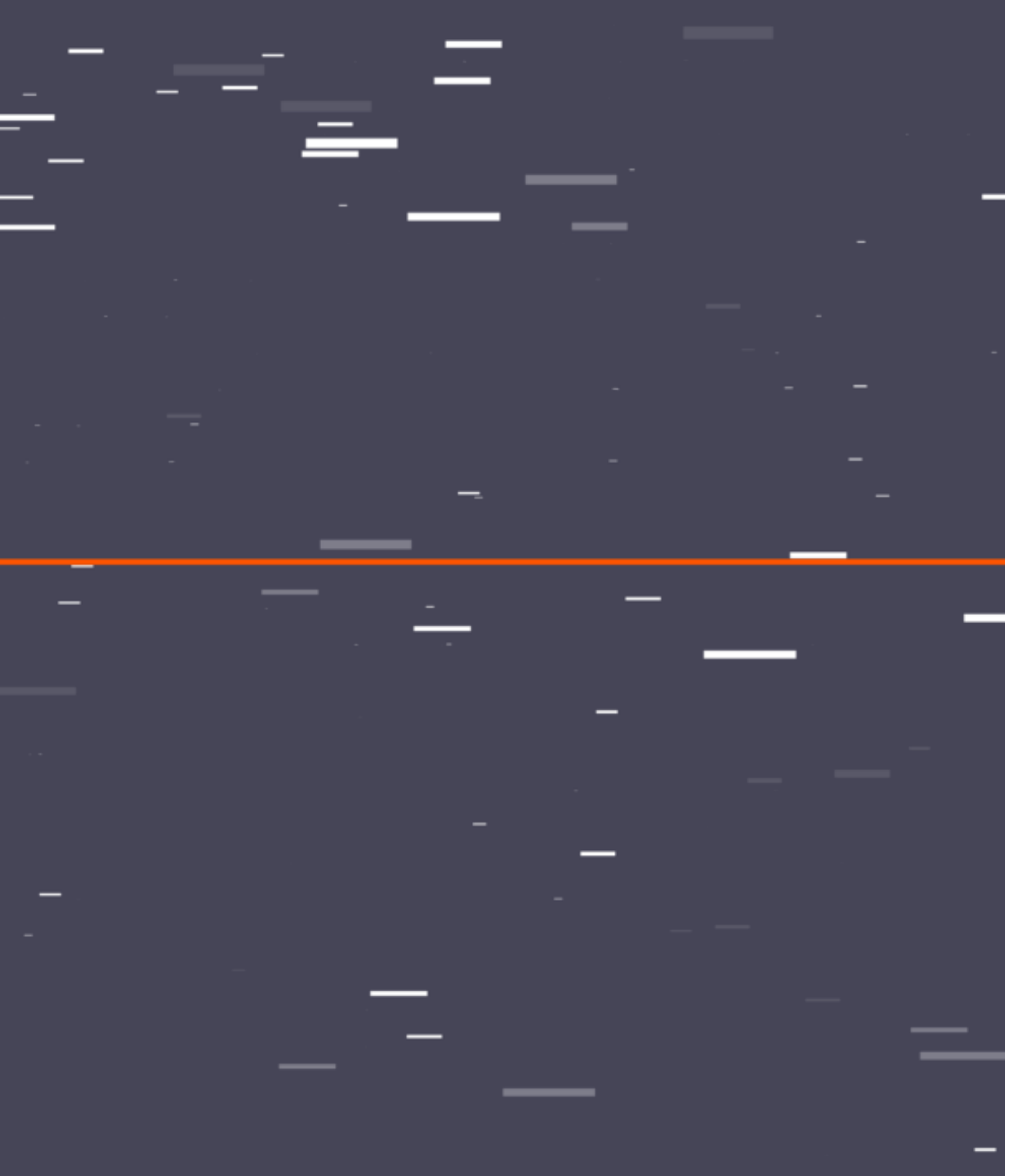
5

Static Signatures

Combines best-in-class signature based anti-virus engines to identify malicious attacks. In addition, we've developed a tool that acts to identify highly complicated signatures.

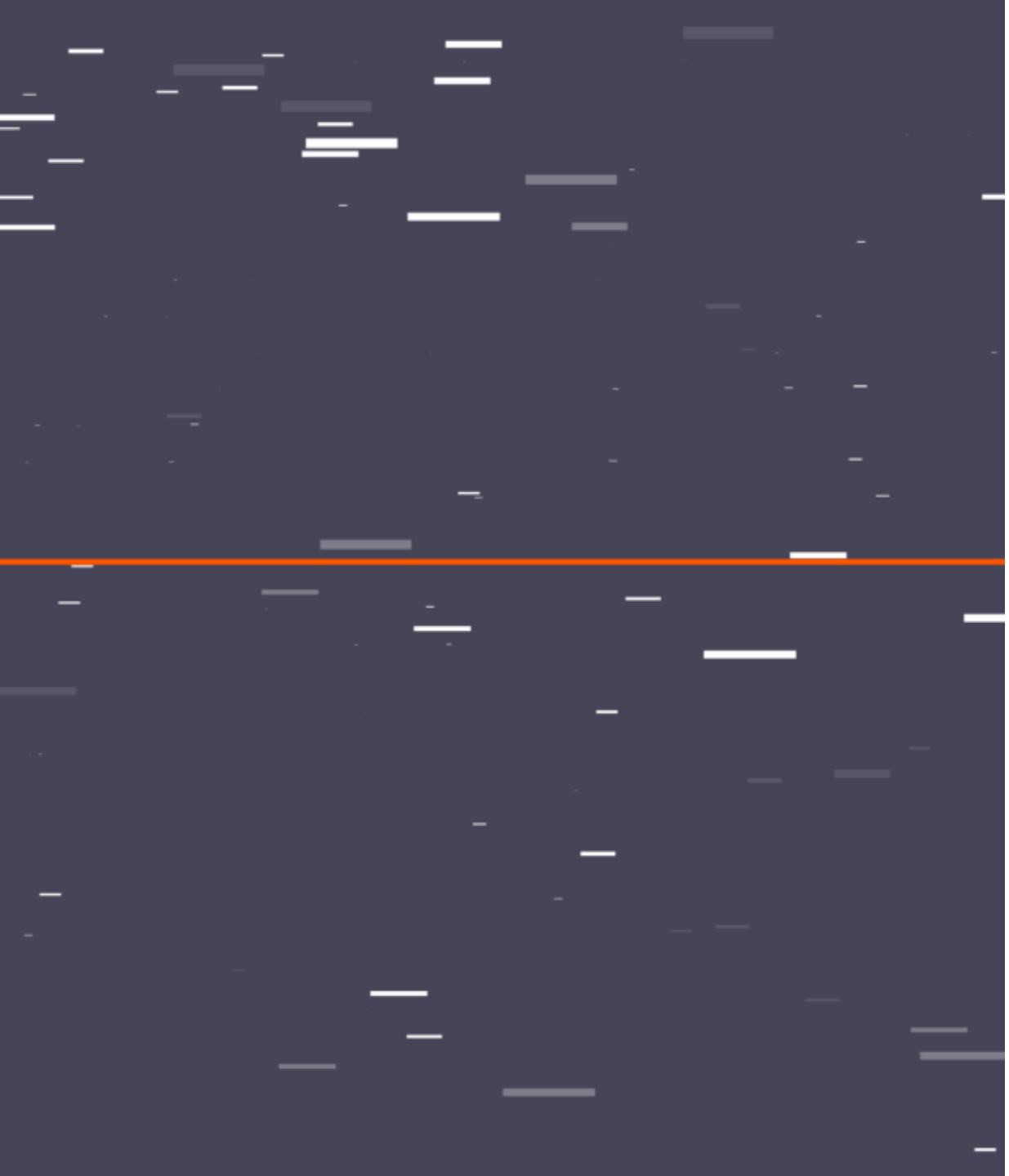


PERCEPTION POINT



Q&A

PERCEPTION POINT



Thank You

PERCEPTION POINT