Size Does Matter

Why Using Gadget-Chain Length to Prevent Code-reuse Attacks is Hard

ENES GÖKTAŞ (VRIJE UNIVERSITEIT AMSTERDAM)

ELIAS ATHANASOPOULOS (FORTH-ICS)

MICHALIS POLYCHRONAKIS (COLUMBIA UNIVERSITY)

HERBERT BOS (VRIJE UNIVERSITEIT AMSTERDAM)

GEORGIOS PORTOKALIDIS (STEVENS INSTITUTE OF TECHNOLOGY)

www.vupen.com/blog/20140520.Advanced_Exploitatior

VUPEN Vulnerability Research Team (VRT) Blog

Advanced Exploitation of Mozilla Firefox Use-After-Free Vulnerability (Pwn2Own 2014)

Published on 2014-05-20 17:19:47 UTC by Arno, Se

Hi everyone,

Pwn2Own 2014 was very exciting a
now getting more secure than ev
however that additional efforts are required to

In this year's edition of Pwn2Own, we have
Internet Explorer 11, Google Chrome, Adob
reported *all* the vulnerabilities and our full e
protect users.

One of the vulnerabilities we have exploi
(MFSA2014-30 / CVE-2014-1512). This flaw w
b
p

blogs.technet.com/b/mmpc/archive/2014/02/17/a-journey-to-cve-2014-0497-exploi

Microsoft

Malware Protection Center

A journey to CVE-2014-0497 exploit

msft-mmpc | 17 Feb 2014 2:50 PM | 1

Last week we published a blog post about a CVE-2013-5330 exploit. We've also recently seen a new, similar attack targeting a patched Adobe Flash Player vulnerability (CVE-2014-0497).

The vulnerability related to this malware was addressed with a patch released by Adobe on February 4, 2014. Flash Player versions 12.0.0.43 and earlier are vulnerable. We analyzed how these attacks work and found the following

ge is visited. When the .swf is loaded, the

able to access an arbitrary location. It overwrites
tion (Note that the exploit does not rely on heap
OP gadgets built from a Flash Player DLL. The
on executable. Finally, the control is passed to the

www.fireeye.com/blog/uncategorized/2014/04/new-zero-day-exploit-targeti

FireEye

New Zero-Day Exploit targeting Internet Explorer Versions 9 through 11 Identified in Targeted Attacks

April 26, 2014 | By Xiaobo Chen, Dan Caselden and Mike Scott | Advanced Malware, Exploits, Targeted Attack, Uncategorized

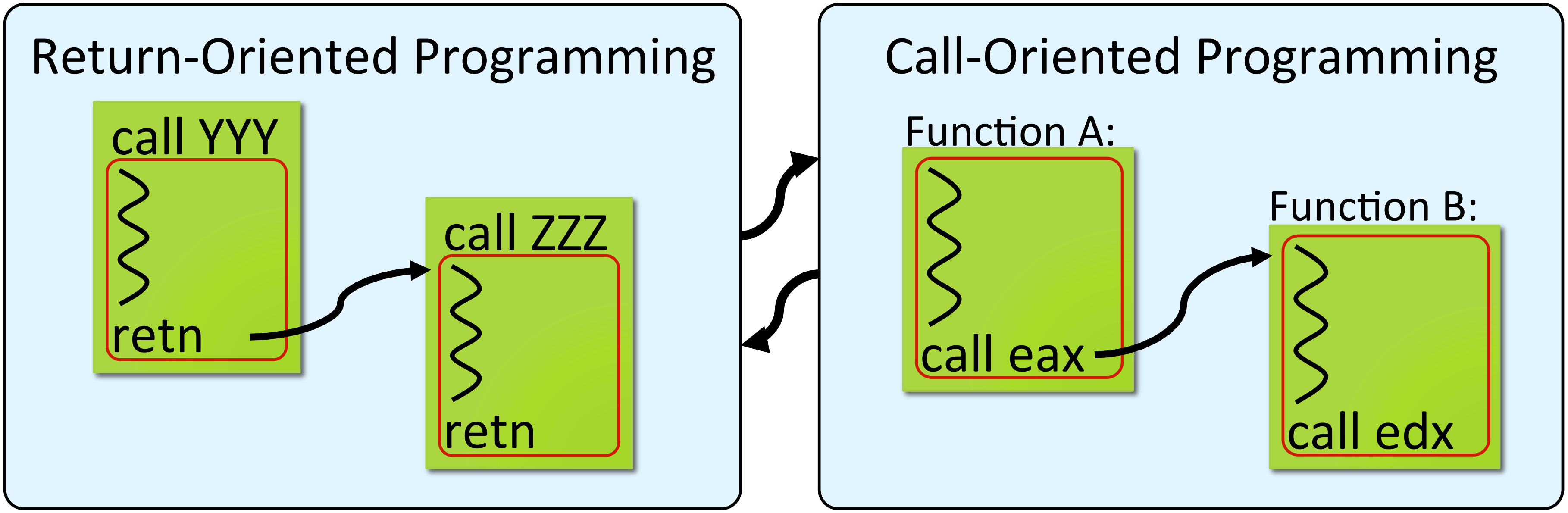
Summary

FireEye Research Labs identified a new Internet Explorer (IE) zero-day exploit used in targeted attacks. The vulnerability affects IE6 through IE11, but the attack is targeting IE9 through IE11. This zero-day bypasses both ASLR and DEP. Microsoft has assigned CVE-2014-1776 to the vulnerability and released security advisory to track this issue.

Control-Flow Integrity

Promising defense mechanism against ROP

We showed that **recent CFI proposals** do not stop ROP attacks
(see "*Out of Control: Overcoming CFI*", Oakland '14)



Inspecting Branching History

Alternative promising defenses against ROP

State-of-the-art proposals:

- kBouncer (Pappas et al., Usenix Security 2013)
- ROPecker (Cheng et al., NDSS 2014)

Assume to be broken

Fundamentally based on:

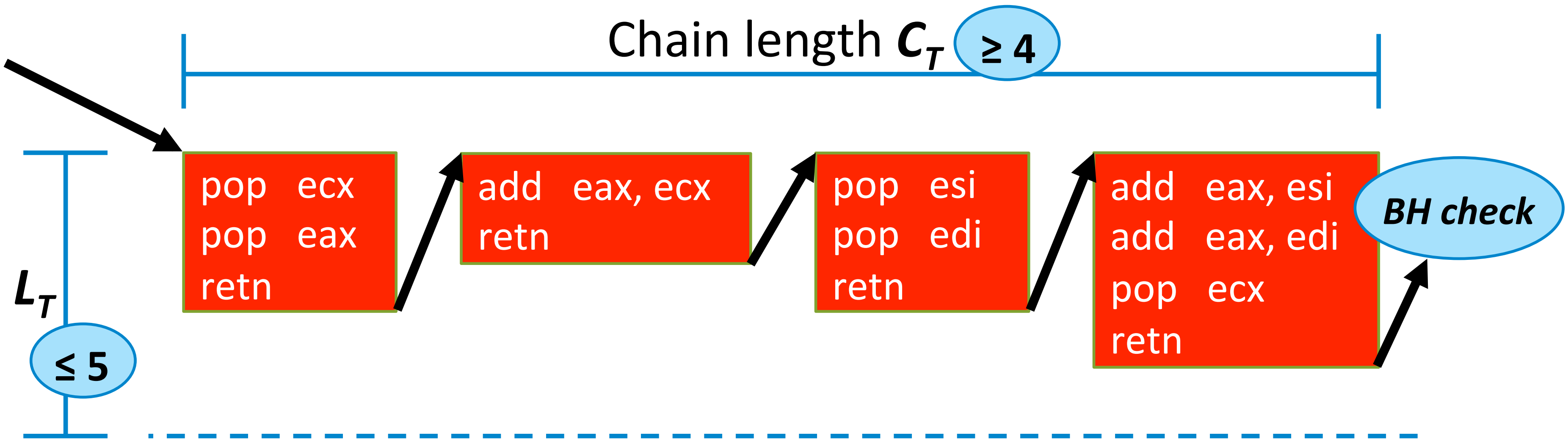
- a **Control-Flow Integrity** policy, and
- a **Heuristic**-based policy

What are the **security implications**?

Focus of this talk

Heuristic-based policy

Relies on **two threshold parameters**

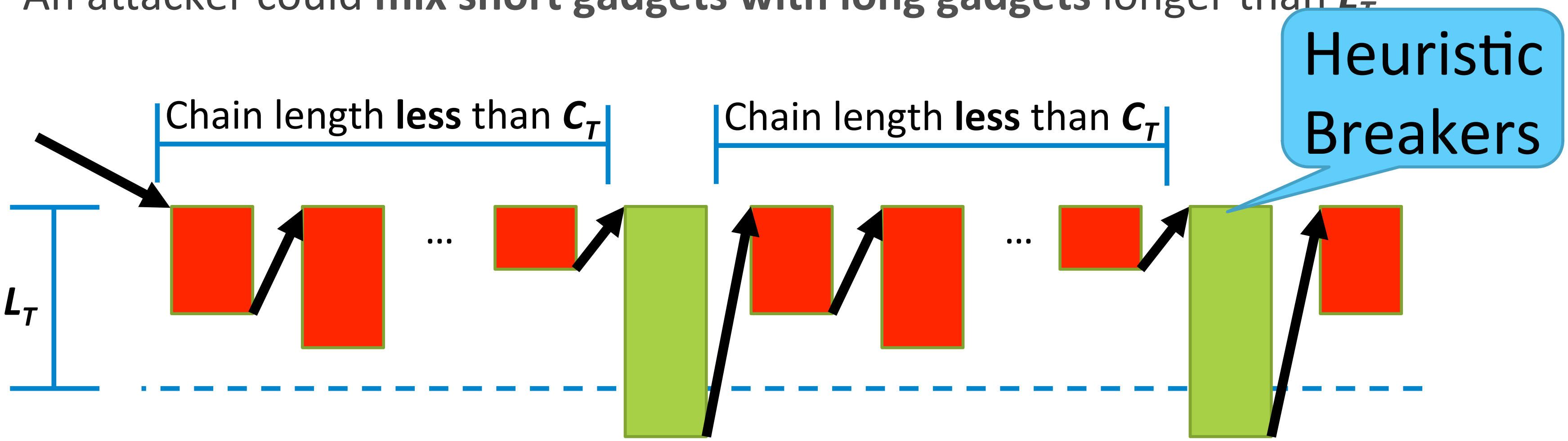


L_T or less number of instructions are considered as **gadgets**
= max gadget length

C_T or more gadgets in sequence is an **attack**
= gadget chain threshold

Picking the "best" Thresholds

An attacker could **mix short gadgets with long gadgets** longer than L_T



Preferably: L_T as large as possible & C_T as small as possible

But setting L_T too large and C_T too small can lead to False Positives

Thresholds have to be chosen carefully!

Chosen thresholds of defenses

	kBouncer	ROPecker
Time-of-Check	Entry of Sensitive API	Entry of Sensitive API + Exit of executable code window
Gadget Length	**20** instructions	**6** instructions
Inspect BH instances	Detected max "benign" gadget chain length: **5**	Detected max "benign" gadget chain length: **10**
Gadget Chain Length	**8** gadgets	**11** gadgets

Difficulties with Heuristic Breakers

Heuristic Breakers may easily:

- Use high number of different registers
- Leave used registers dirty at exit
- Require memory preparations
- Have a whacky code sequence

```
mov eax, ebx
mov ecx, edx
add esi, edi

mov esi, [0x1234]
cmp esi, 10
jg  X

mov ecx, 0x2321
div ecx
mov [eax], edi

mov ecx, 0x5678
and edi, ecx
xor eax, edi
retn
```

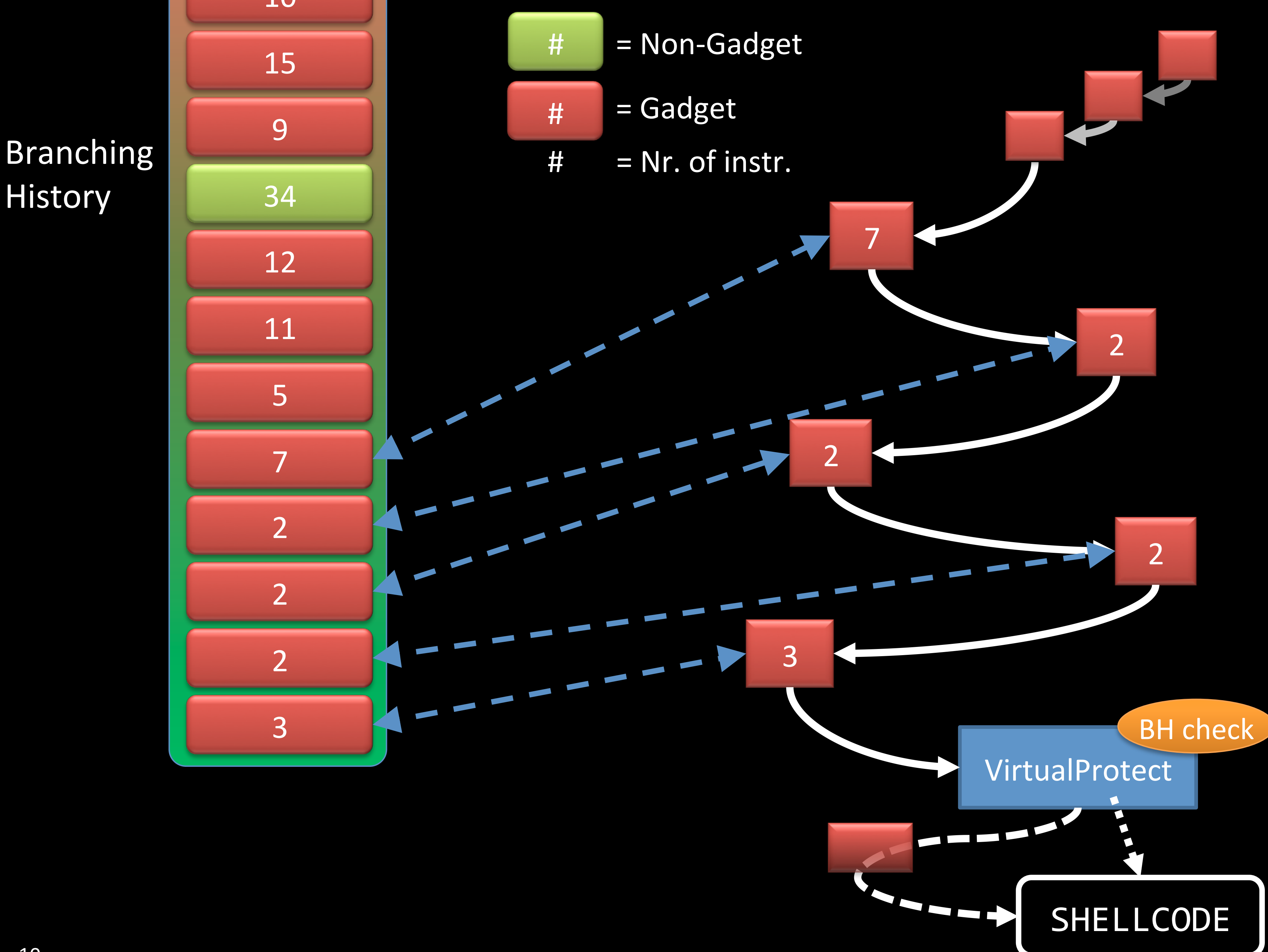
Proof-of-Concept Exploit

Real IE8 vulnerability

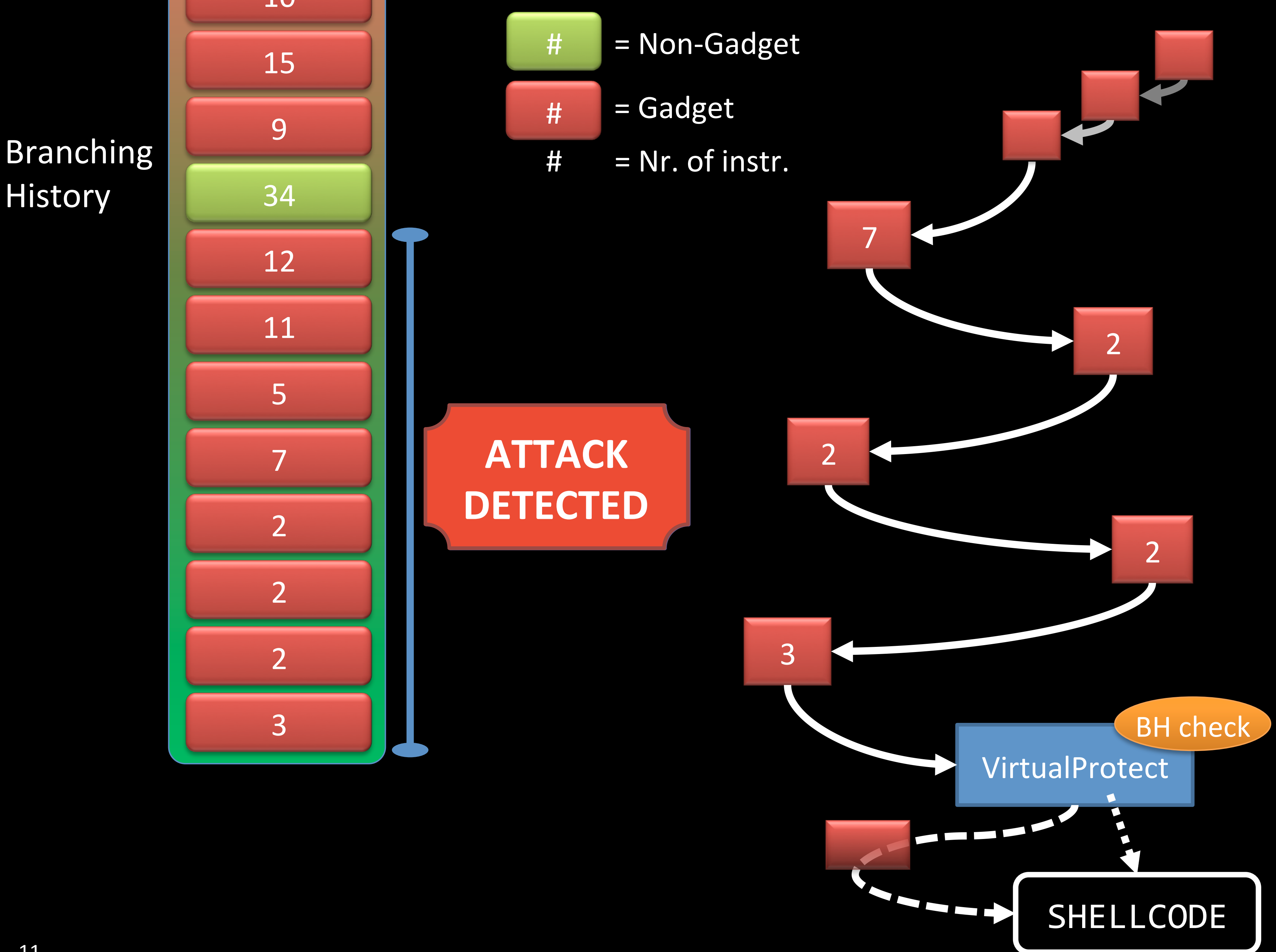
Bypasses ASLR, DEP, kBouncer

Idea: intersperse a Heuristic breaker in ROP chain to prevent reaching $\boldsymbol{C_T}$

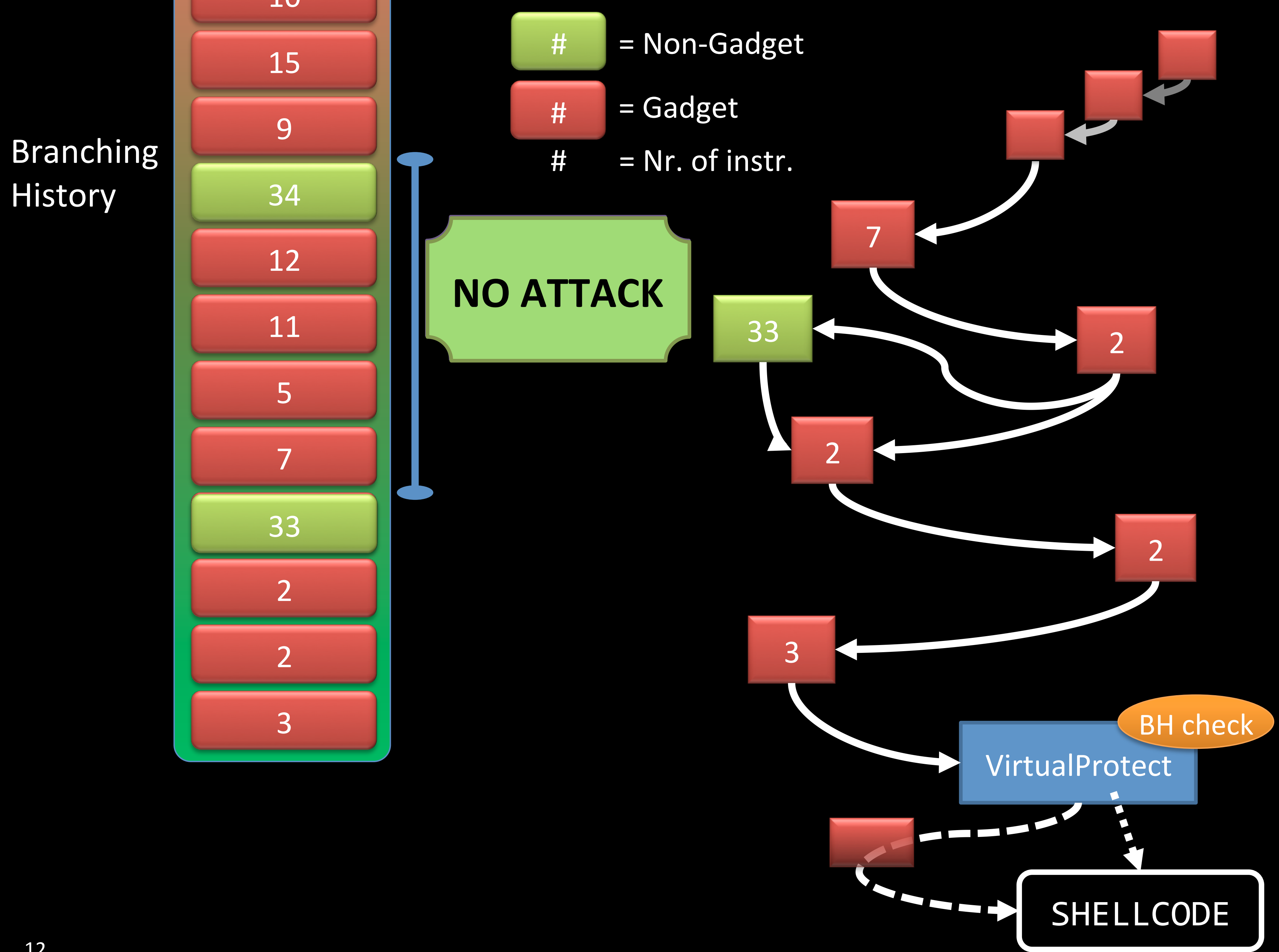
Goal: execute our injected code



Branching
History
15
9
34
12
11
5
7
2
2
2
3
= Non-Gadget
= Gadget
= Nr. of instr.
7
2
2
2
3
BH check
VirtualProtect
SHELLCODE



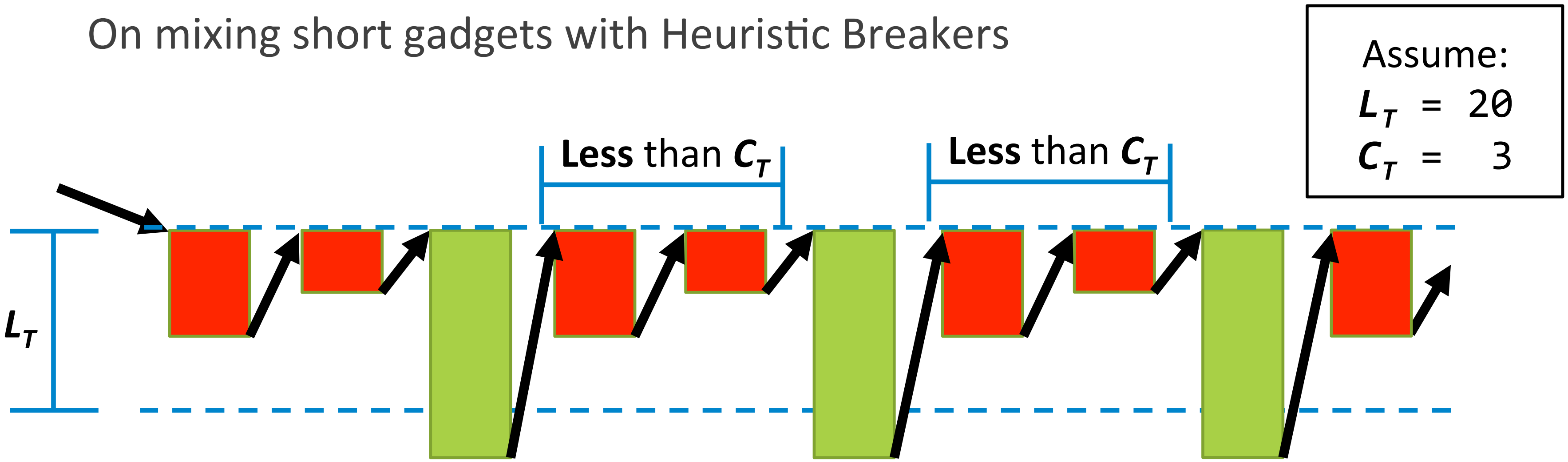
= Non-Gadget
= Gadget
= Nr. of instr.
Branching History
15
9
34
12
11
5
7
2
2
2
3
ATTACK DETECTED
7
2
2
2
3
BH check
VirtualProtect
SHELLCODE



Branching History
16
15
9
34
12
11
5
7
33
2
2
3
= Non-Gadget
= Gadget
= Nr. of instr.
NO ATTACK
7
33
2
2
2
3
BH check
VirtualProtect
SHELLCODE

Implications of Stricter Thresholds

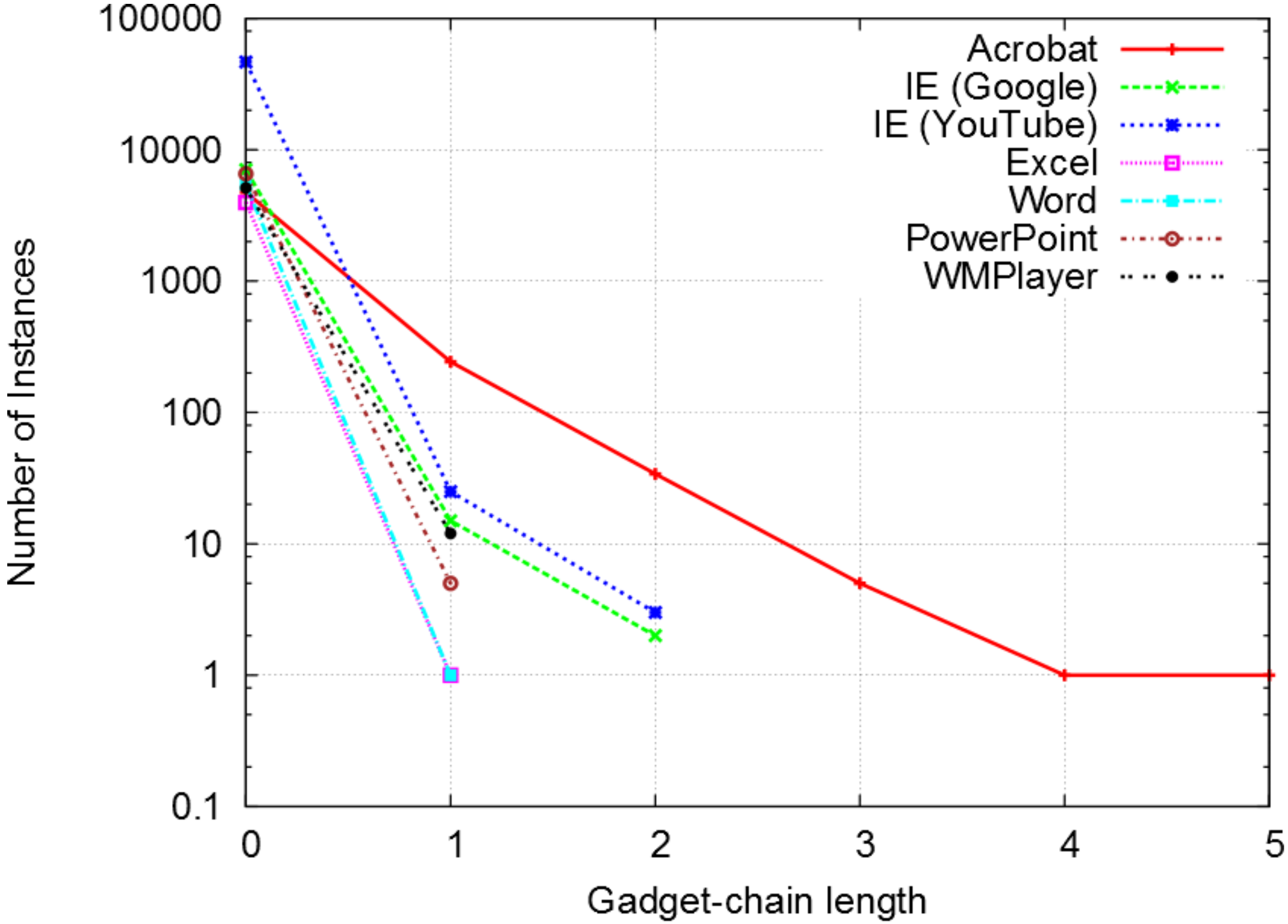
On mixing short gadgets with Heuristic Breakers



Difficulties for an attacker:

- Not enough space to **prepare Heuristic Breaker**
- Not enough space to **restore state after Heuristic Breaker**
- Not enough space to **prepare a function call**

Per Application Thresholds



Conclusion

Choosing the right thresholds for ROP detection is difficult

The “long gadgets are not usable” assumption is broken

We need better tools to evaluate our defenses