



# Advanced Persistent Response

Peleus Uhley | Platform Security Strategist



# Outline

- Background
- Analyzing threat
- Finding resources
- Planning & executing a response
- Conclusion

“usually refers to a group with both the capability and the intent to persistently and effectively target a specific entity”

- Wikipedia

# Flash Player CVEs in the wild



- ◆ 4 SWF in PDF attacks
- ◆ 5 SWF in Office document attacks
- ◆ 3 XSS attacks
- ◆ 1 protocol based attack
- ◆ 2 attacks after patch released

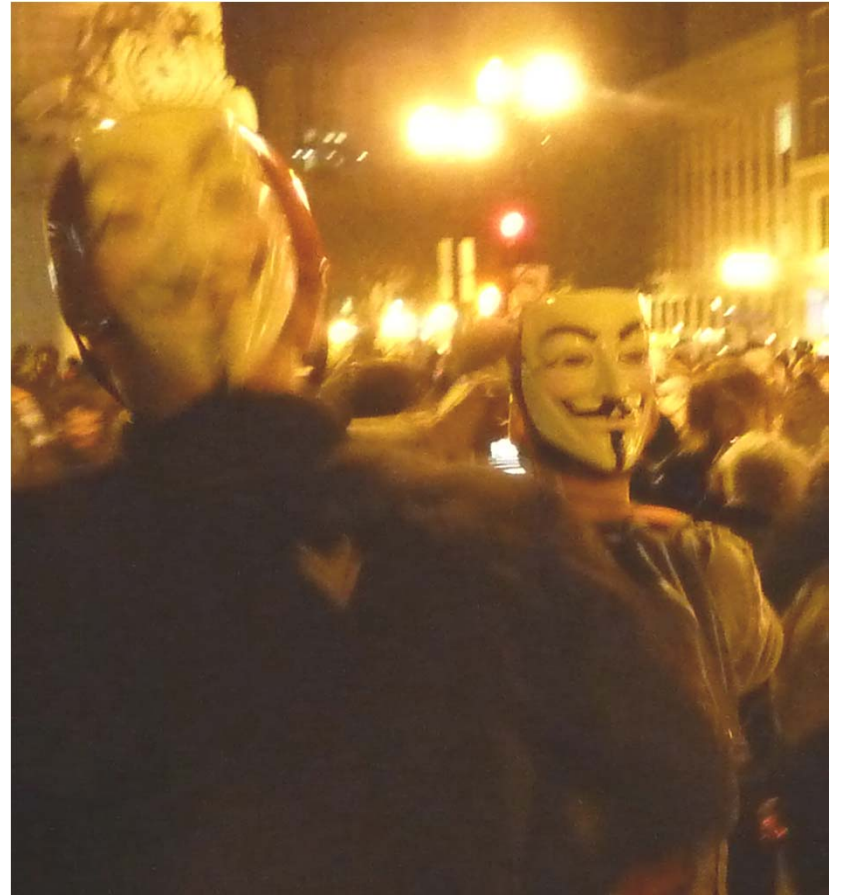
The strategy necessary to inhibit or reduce a malicious entity or entities' capabilities to repeatedly conduct attacks leveraging a specific platform or against a specific target.

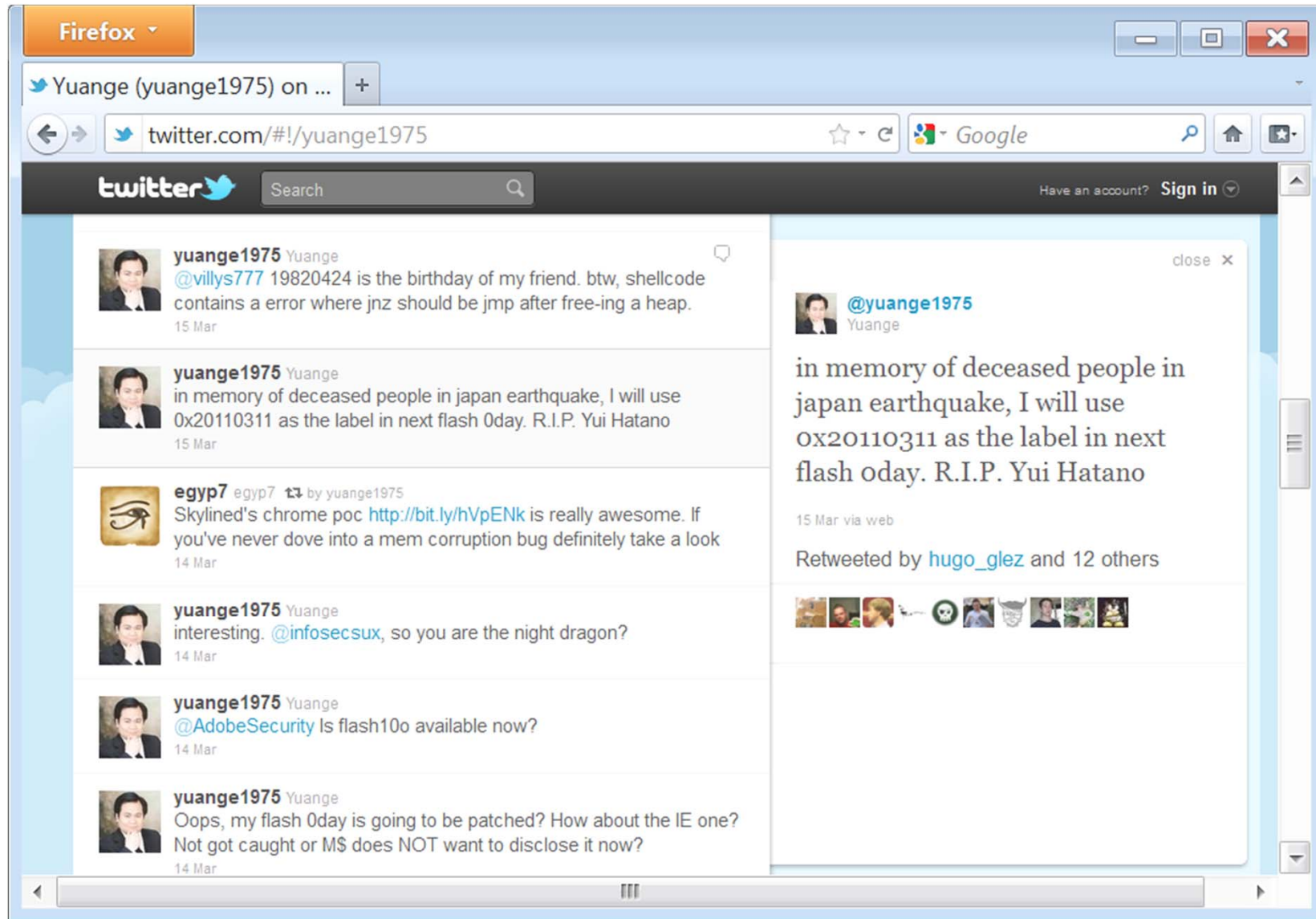
# Information Gathering



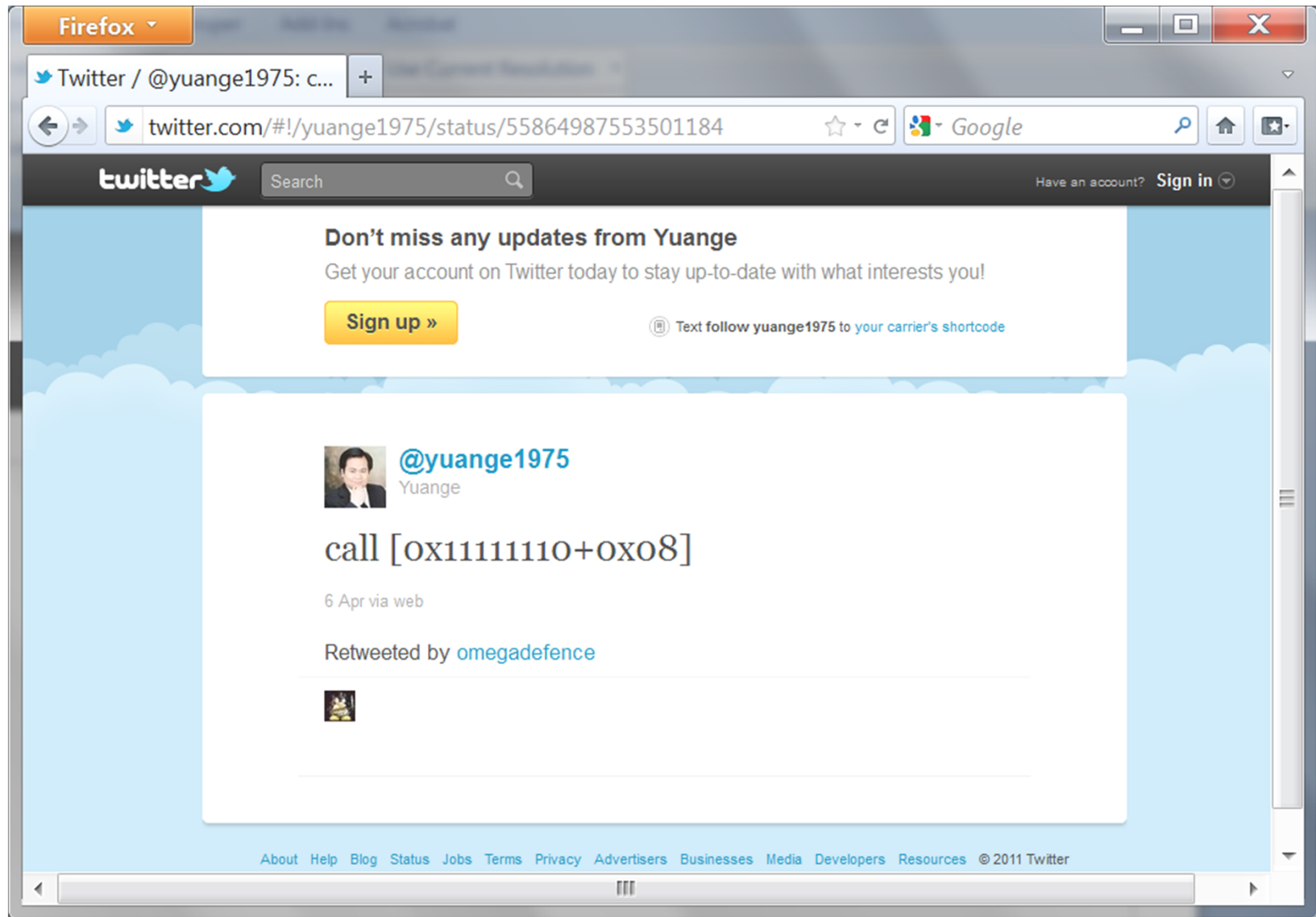
# Know your adversary

- Who are they?
- Why are they upset?
- Does it change over time?









# Krebs on Security

In-depth security news and investigation



ABOUT THIS BLOG

## Posts Tagged: Yuange1975

A Little Sunshine / The Coming Storm — 21 Comments

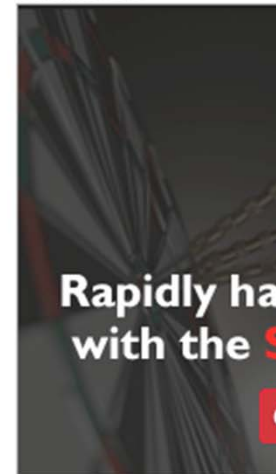
### 3 **Advanced Persistent Tweets: Zero-Day in 140 Characters**

MAY 11

133 tweets  
TOP ★1K  
retweet

The unceasing barrage of targeted email attacks that leverage zero-day software flaws to steal sensitive information from businesses and the U.S. government often are described as being ultra-sophisticated, almost ninja-like in stealth and anonymity. But according to expert analysis of several recent zero-day attacks – including the much publicized break-in at security giant RSA – the Chinese developers of those attack tools left clues aplenty about their identities and locations, with one apparent contender even Tweeting about having newly discovered a vulnerability days in advance of its use in the wild.

Advertisement




Recent Posts


# Beware of false positives


## Results for 0day flash


Tip: use [operators](#) for [advanced search](#).


**Tweets**   [Tweets with links](#)   [People](#)





**asintsov** Alexey Sintsov   
is that a **0day Flash** bug in the wild on MAIL.RU with LG bannner??  
<http://twitpic.com/4wjcha>  
4 hours ago  
[Top Tweet](#)






**toucansystem** Toucan System   
RT @asintsov: is that a **0day Flash** bug in the wild on MAIL.RU with  
LG bannner?? <http://twitpic.com/4wjcha>  
1 hour ago





**Geekpirat** Geekpirat   
RT @asintsov: is that a **0day Flash** bug in the wild on MAIL.RU with  
LG bannner?? <http://twitpic.com/4wjcha>  
2 hours ago




**sh2kerr** Alexander Polyakov    
rt: @asintsov is that a **0day Flash** bug in the wild on MAIL.RU with  
LG bannner?? <http://twitpic.com/4wjcha>  
2 hours ago



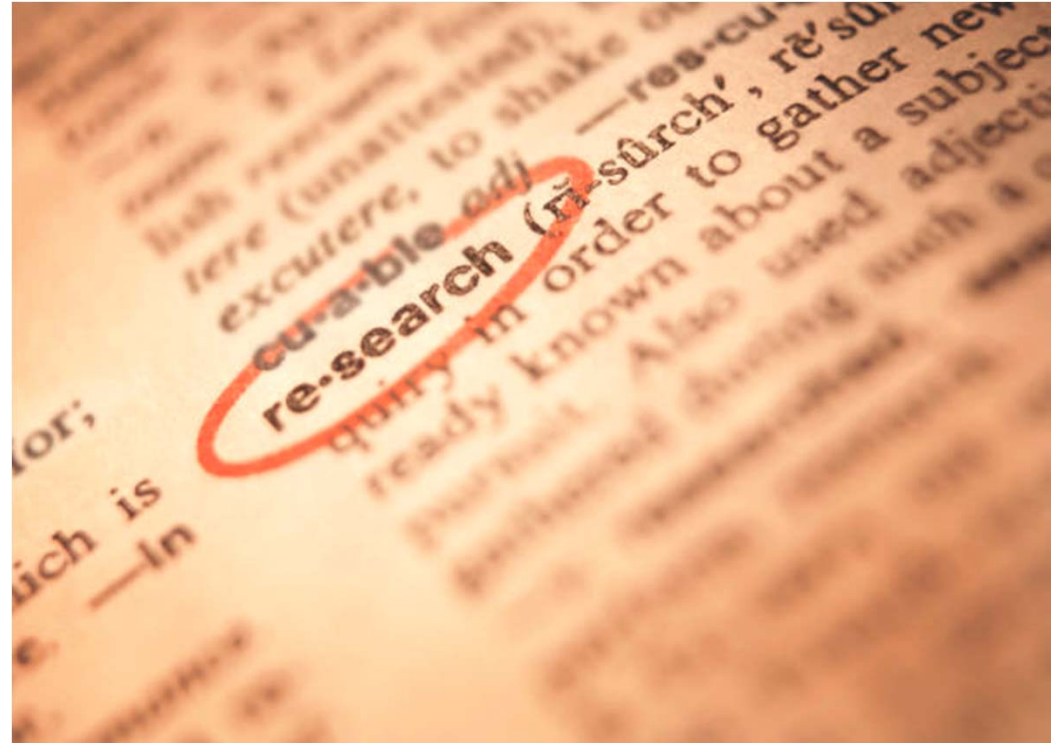
**endrazine** Jonathan Brossard   
RT @asintsov: is that a **0day Flash** bug in the wild on MAIL.RU with  
LG bannner?? <http://twitpic.com/4wjcha>  
3 hours ago



**ChristiaanBeek** Christiaan Beek   
is that a **0day Flash** bug in the wild on MAIL.RU with LG bannner??  
<http://twitpic.com/4wjcha> (via @asintsov)  
3 hours ago

# Technical Analysis

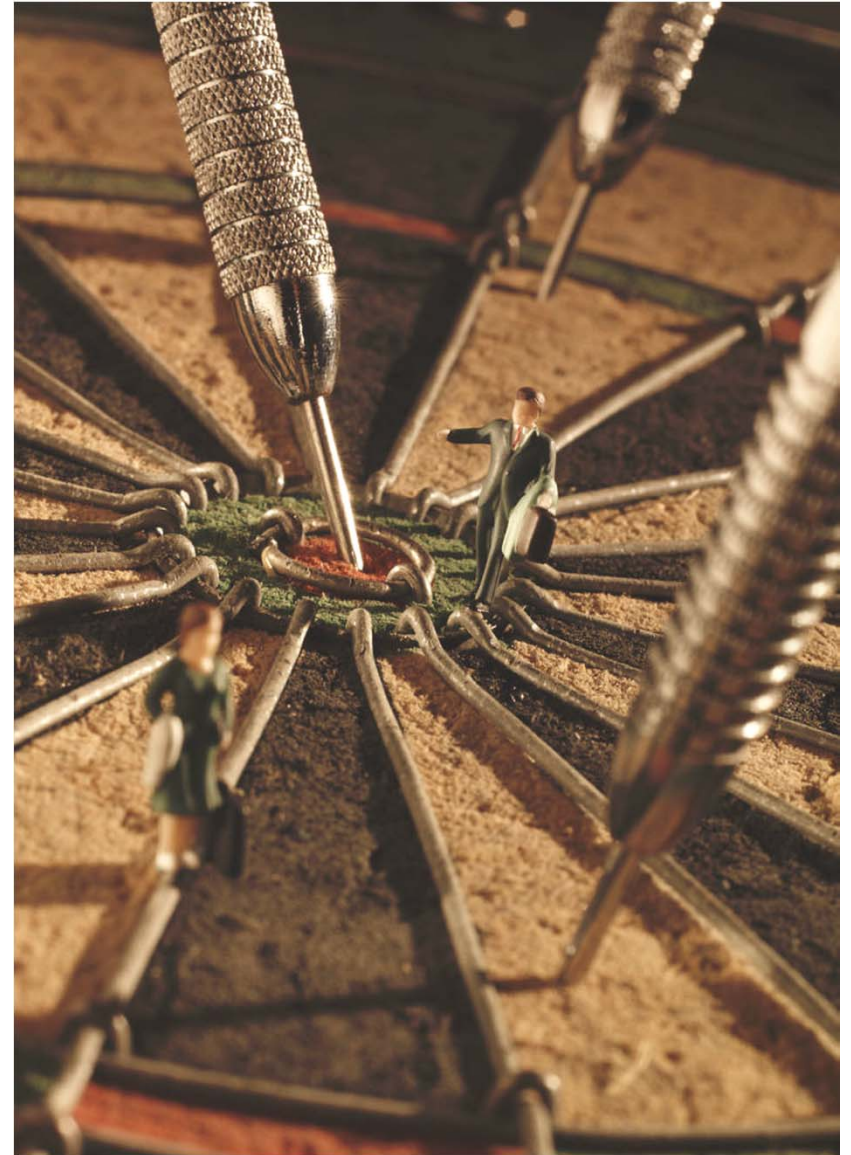
- Their sources?
- Their targets?
- Their skill?



- Series of attacks based on SWFs from flashandmath.com
- Indicates the areas of code that are being attacked
- Learn from mimicking their approach

# Targets

- Reports from government customers
- Document-based spear phishing attacks
- Most exploits are never widely deployed



- Symantec studied payloads from 2006 -> 2011
- Attacks grouped based on the malware installed (Sykipot)
- Large command & control botnet
- Mostly used zero-day attacks within several different products to install

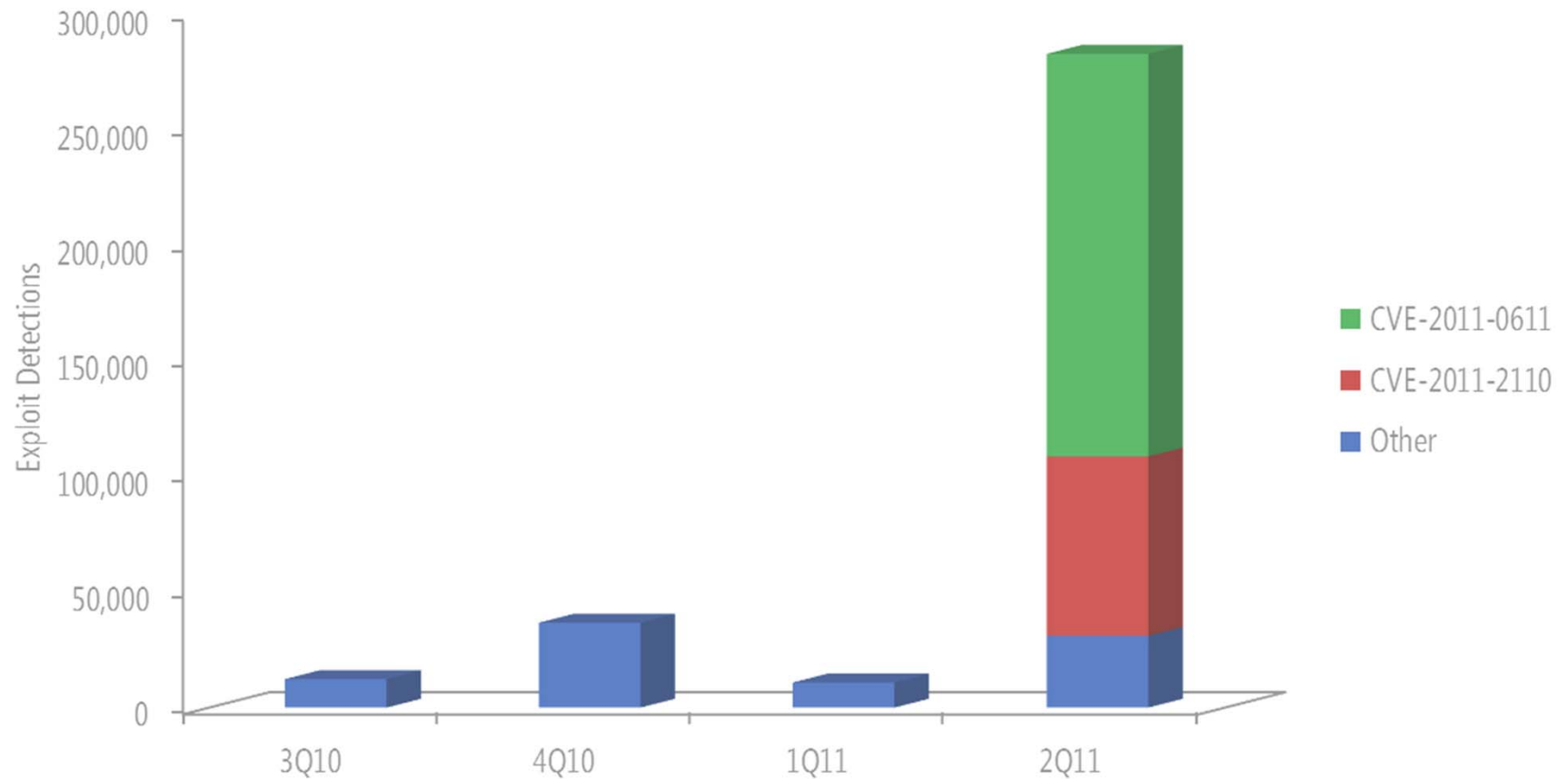
“Thus, the Sykipot attackers are likely to be an organized and skilled group of individuals. Given their persistence and their long-running campaigns, the attackers are likely to have consistent funding for their efforts.”

- Symantec Blog, December 08, 2011



- A change in targets will require changes to security feature strategy
- Changes may correspond to specific events
- Example:
  - Exploit kits started using SWFs in early 2011
  - Used attacks that were at least 2 months old

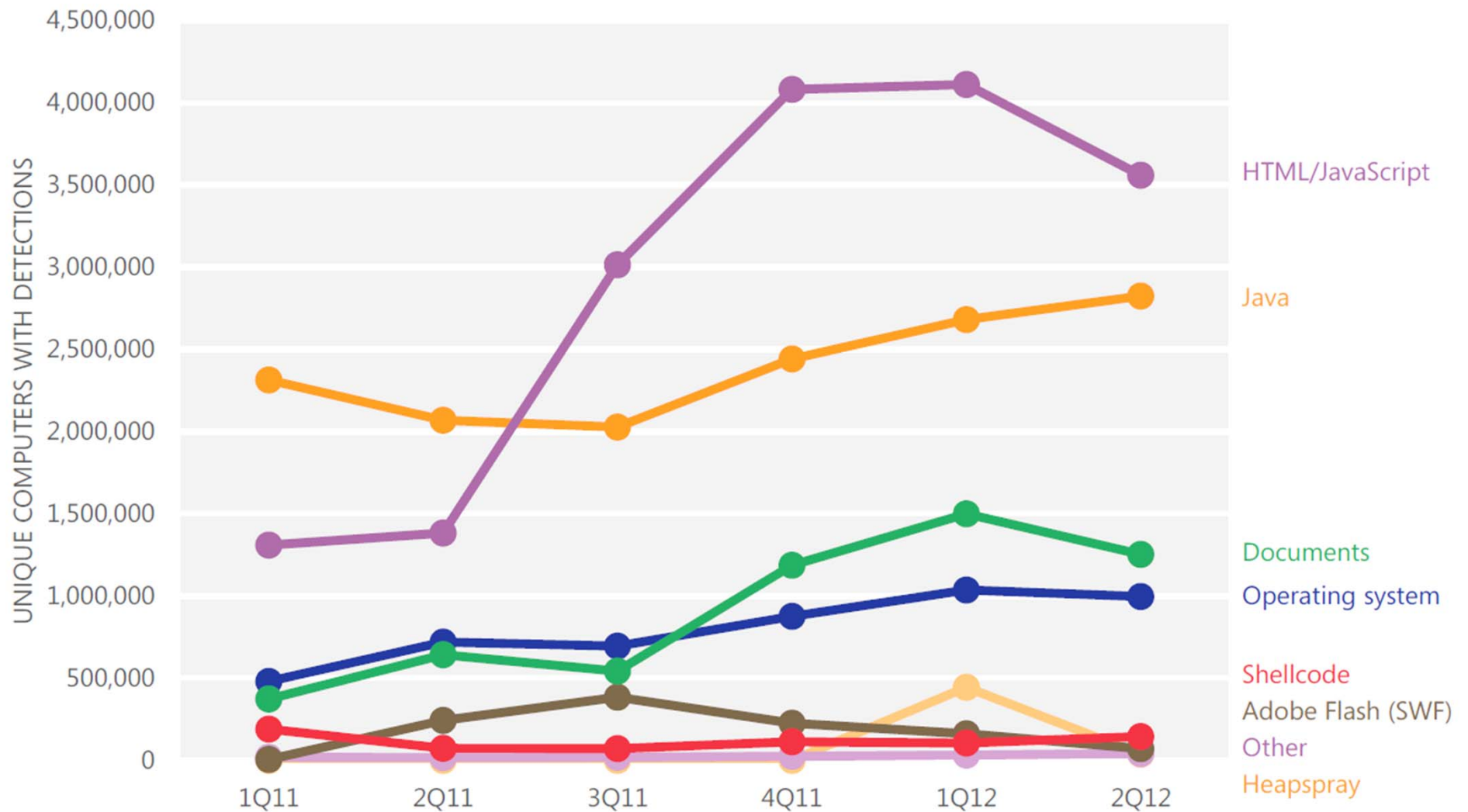
# Effect of exploit kits



Source: Microsoft Security Intelligence Report Volume 11

# Everything is relative, everything changes

Figure 9. Unique computers reporting different types of exploits, 1Q11–2Q12



Source: Microsoft Software Intelligence Report Volume 13

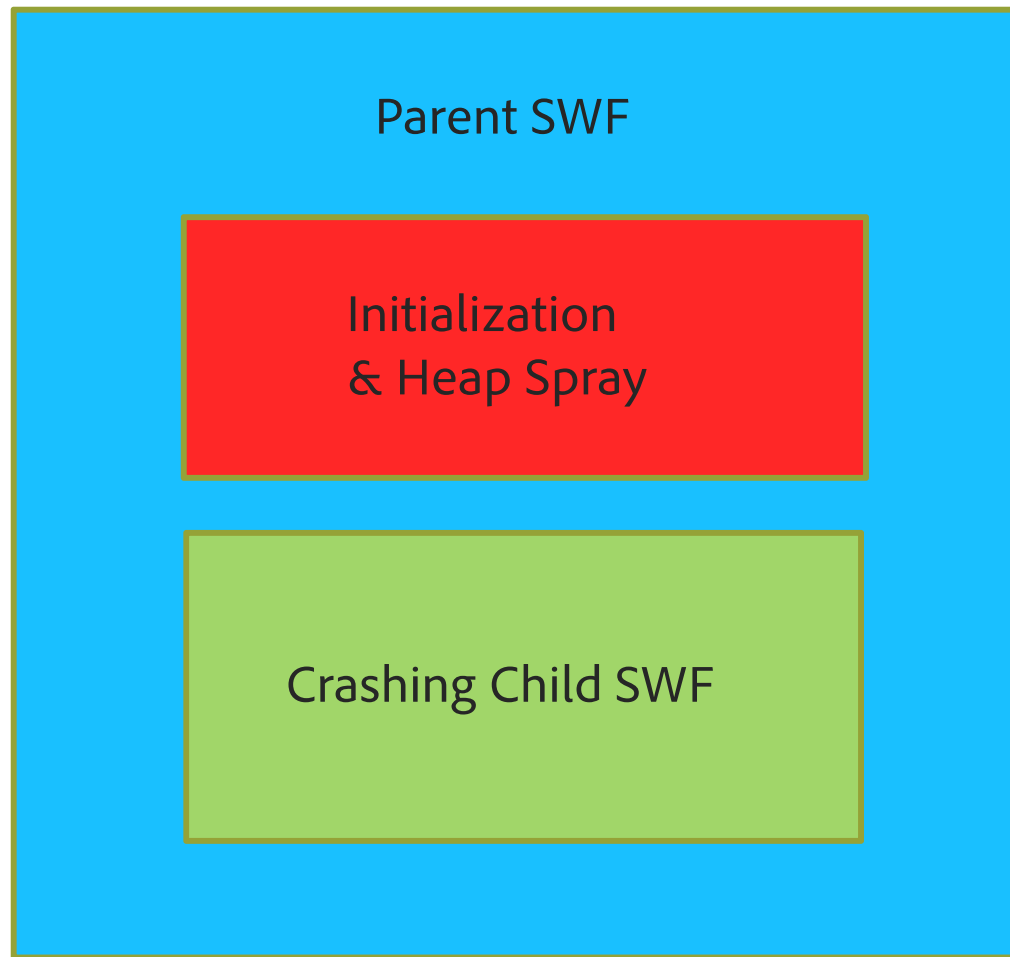
# Evolution of attacks

- Attackers gain skill with practice
- Hackers will utilize published research
- Makes signature development more difficult



- Simple bit flip of existing SWF from the web
- A second SWF was used for the heap spray
- Both SWFs were inside a PDF
- Flash was a means to an end

- Eventually moved to a single file approach





- Dynamically passing obfuscated data

```
main.swf?info=02E6B1525353CAA8AD555555AD31B3D73034B657AA31B4  
B5AFB5B2B537AF55543549AEB550AC55303736B337AF51D3527B7AF4C66  
B7E
```

- Targeting specific versions

```
if ((((((Capabilities.version.toLowerCase() == "win 10,3,181,14")) ||  
((Capabilities.version.toLowerCase() == "win 10,3,181,22")))) ||  
((Capabilities.version.toLowerCase() == "win 10,3,181,23"))))){
```

- Return orientated programming



## Conversion to logic based attacks

- CVE-2011-2107, CVE-2011-2444 & CVE-2012-0757 were XSS attacks
- Required ActionScript programming knowledge
- Trial & error methods used to identify vulnerabilities
- Used SWF obfuscation tools

- Protocol based bug that required an RTMP server
- Faster deployments, more overlaps
- Used as a tool for bugs in other products (Elderwood gang)

# Overall advancements

2009

- Dumb fuzzing
- Brute force

2010

- Modularizing code
- Experimenting with new research

2011

- ROP exploitation
- Bypassing mitigation strategies
- Full language understanding

2012

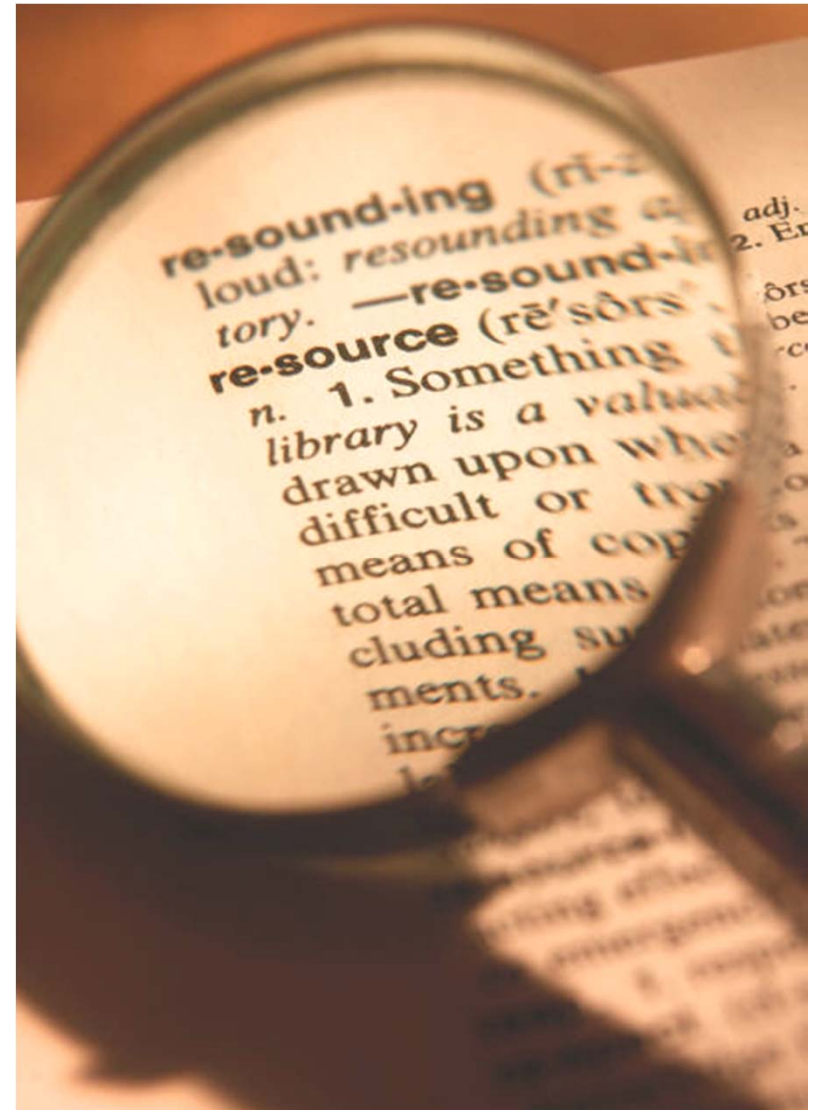
- Expanding bug finding techniques
- Flash Player used to assist in exploiting bugs in other products

# Resourcing a response plan



# All you need is...

- More time
- More money
- More hardware
- More people
- More...

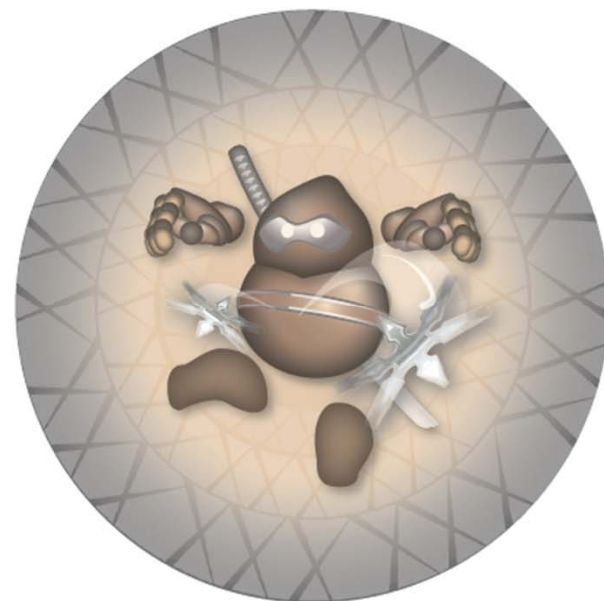


# Balancing the load

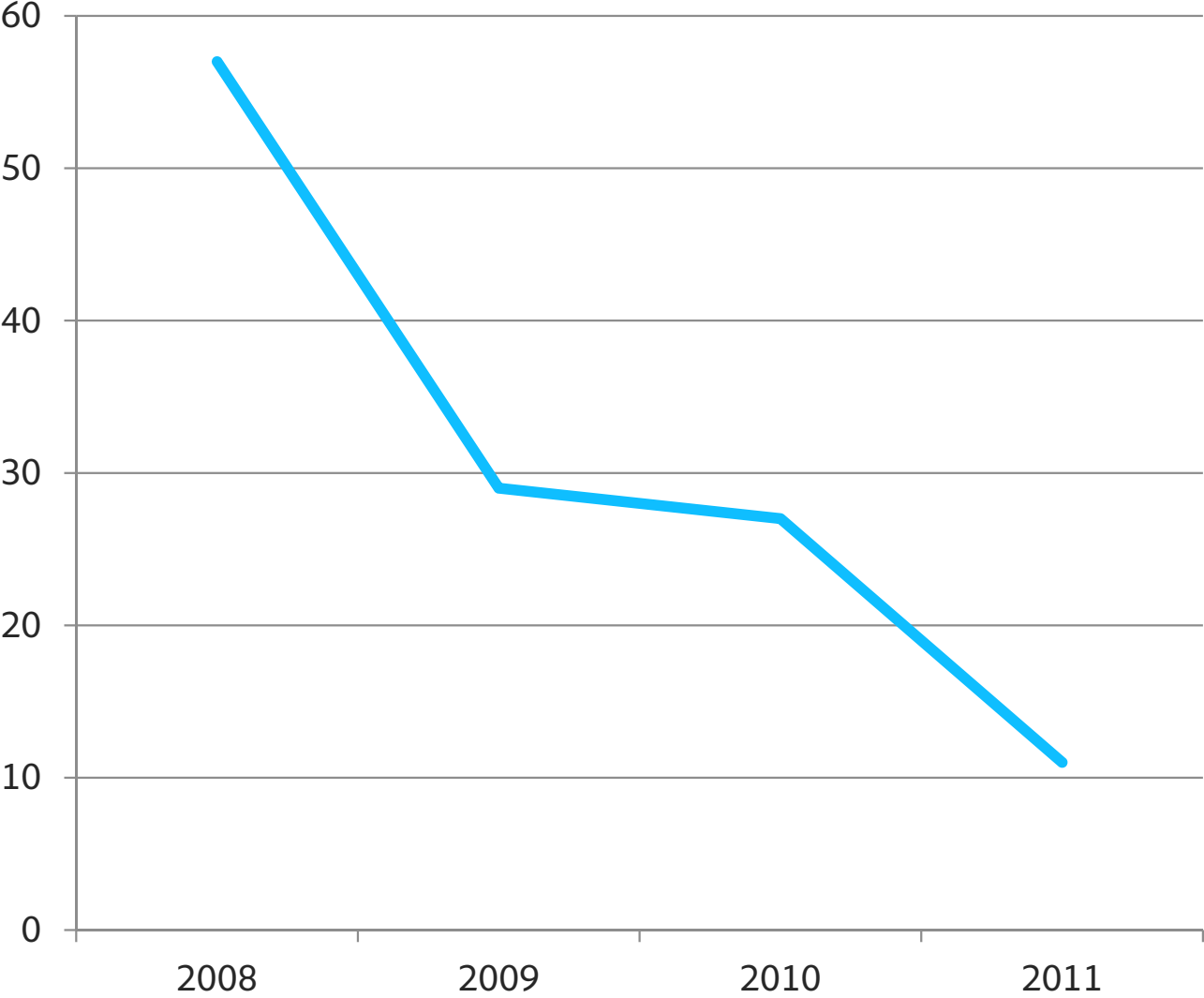
- Security teams should focus on larger, high value projects (“security features”)
- Developers work on security bugs



- Minimum security training level for everyone
- Everyone within the Flash Runtime team has at least a white belt security certification
- Brown belt projects allow the entire company to assist!



# Adobe training results



— Zero Day Response Over Time



Make friends!

“It is easy enough to be friendly to one's friends. But to befriend the one who regards himself as your enemy is the quintessence of true religion. The other is mere business.”

— Mahatma Gandhi



# Types of friends

- Researchers
- Business partners
- Defensive software companies
- Victims of attacks
- Tool vendors



# Planning a response strategy



# Secure Product Lifecycle?



1. Increase the difficulty of exploitation.
2. Limit the window of opportunity for use.

# Killing bugs



VS



## Fuzzing at scale (Round 1)

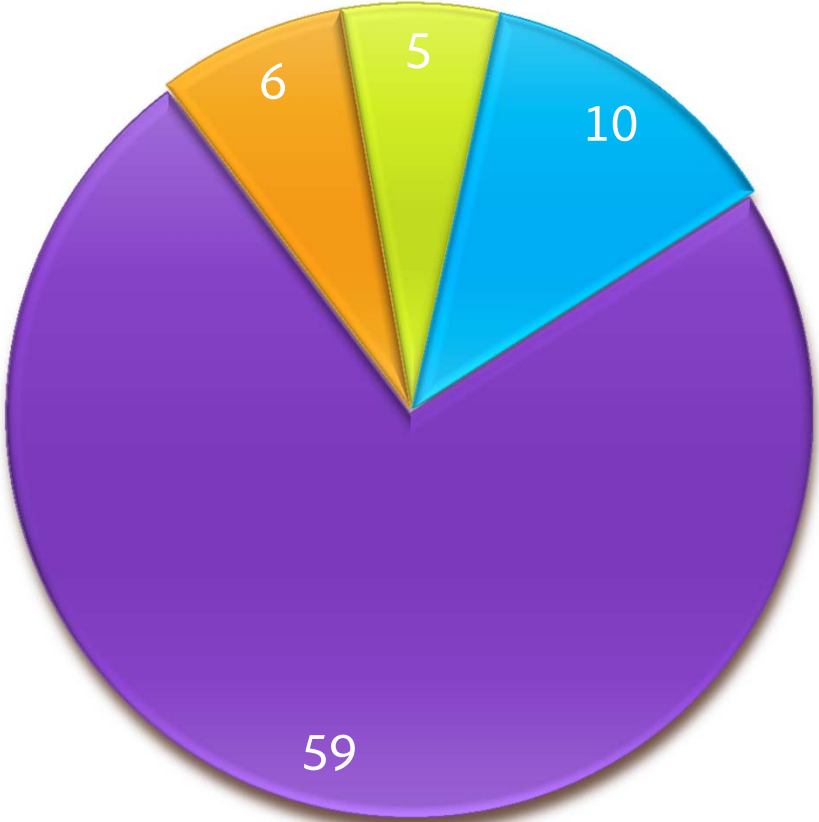
- Partnered with Google on FP fuzzing effort
- 2,000 CPUs
- Corpus distillation of 2 TB of SWFs into 20,000 files (1 week)
- 3 weeks of fuzzing
- Bit flipping approach
- 1 fuzzing guru (Tavis Ormandy)

## “Patching at scale”

- 400 files distilled into 80 unique issues
- Used fuzzers to reproduce and classify issues
- Authored app to auto-file bugs
- Created tiger team to address the issues
- Majority of issues addressed within 60 days



**!exploitable**



- Exploitable
- Probably Exploitable
- Unknown
- Probably Not Exploitable

## Alternative measurements (Round 1)

“This is completely unfair competition and unfair practices vis-a-vis other security researchers (or fuzzer enthus).

...

You guyz killed couple of my bugs.”

- TestFuzzer, August 16, 2011

## Alternative measurements (Round 2)

“Just checked, 0day in Flash which I prepared for #pwnium2 has been killed. Thank You very much @j00ru @fjserna You are real miscreants!”

-Nikita Tarakanov, October 8, 2012

## Lessons learned

- Bugs were spread across the entire code base
- Eliminated some low hanging fruit
- 1 code change per 12,600 CPU hours (1.44 years)

## Using fuzzing as hints

- 1 good dev == CPU years of fuzzing effort
- Fuzzing can provide hints of where to focus code review.
- Focus on code cleanup rather than bug fixing.

And, of course...



we are releasing new exploit with  
Vulndisco Step-Ahead: Flash Player Oday,  
bypasses DEP/ASLR and works with  
FF,IE, Chrome

21 hours ago via web

Retweeted by [n0p](#) and 15 others

There is always one more to  
find...

# Integrating security defenses



# Holistic mitigations

- Work smarter, not harder.
- More effective at deterring attacks
- Require experienced resources
- Require longer periods of development time.

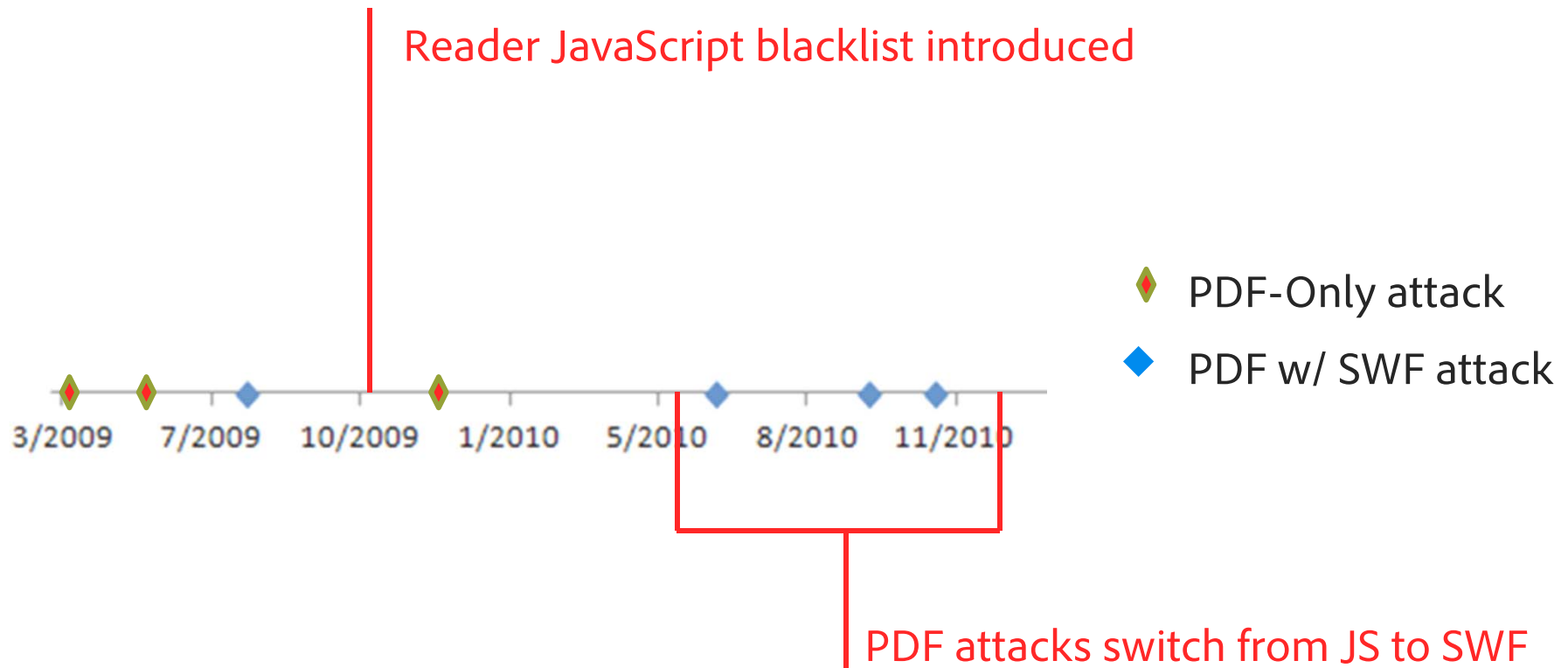


# Adobe Reader examples

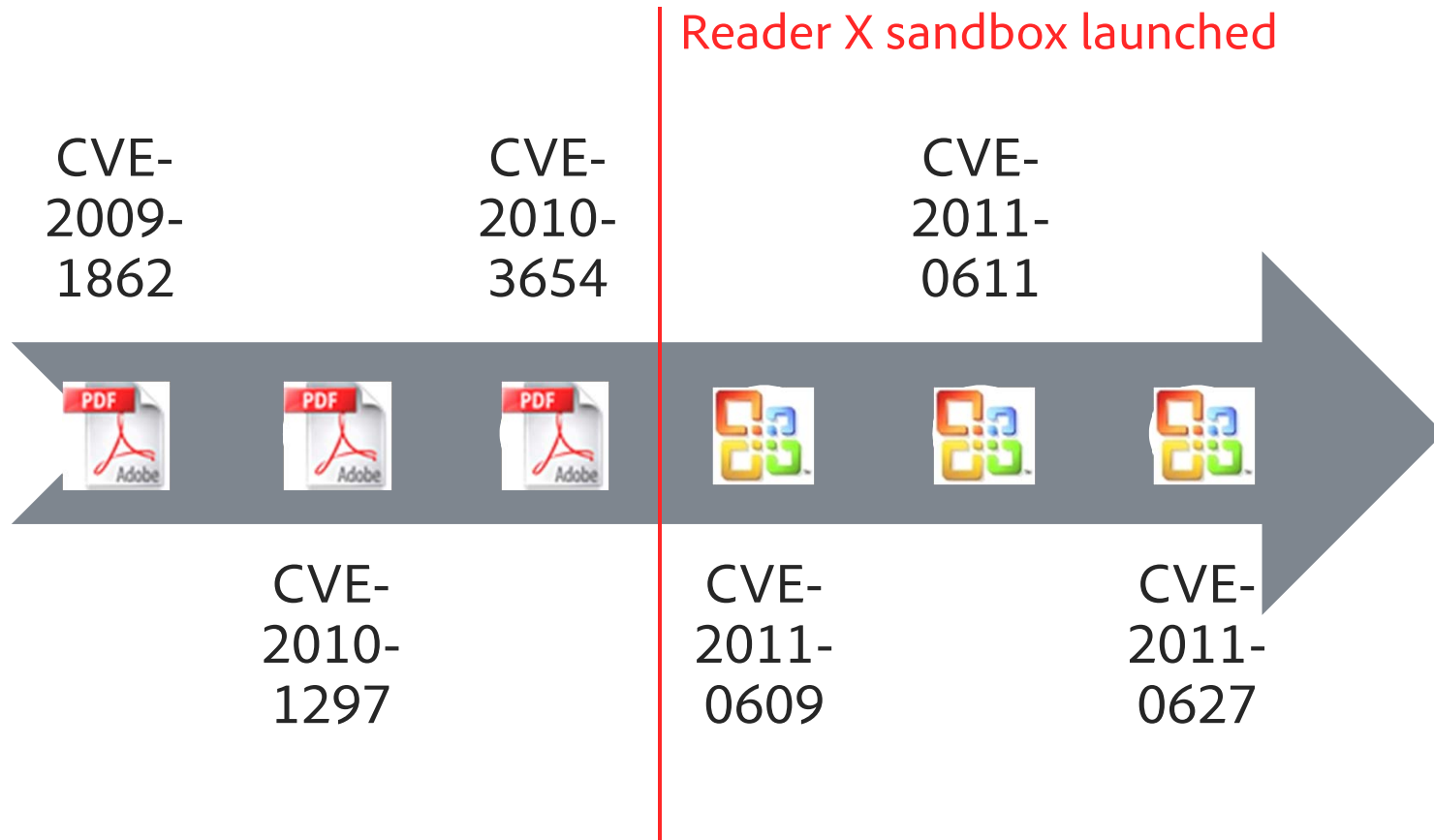
- JavaScript blacklist
- Improved updater
- Reader X Sandbox
  - Dedicated team for over 1 year of effort
  - Additional engineers for misc. support
  - External consultants



# Cause and effect



# Effect of sandboxing



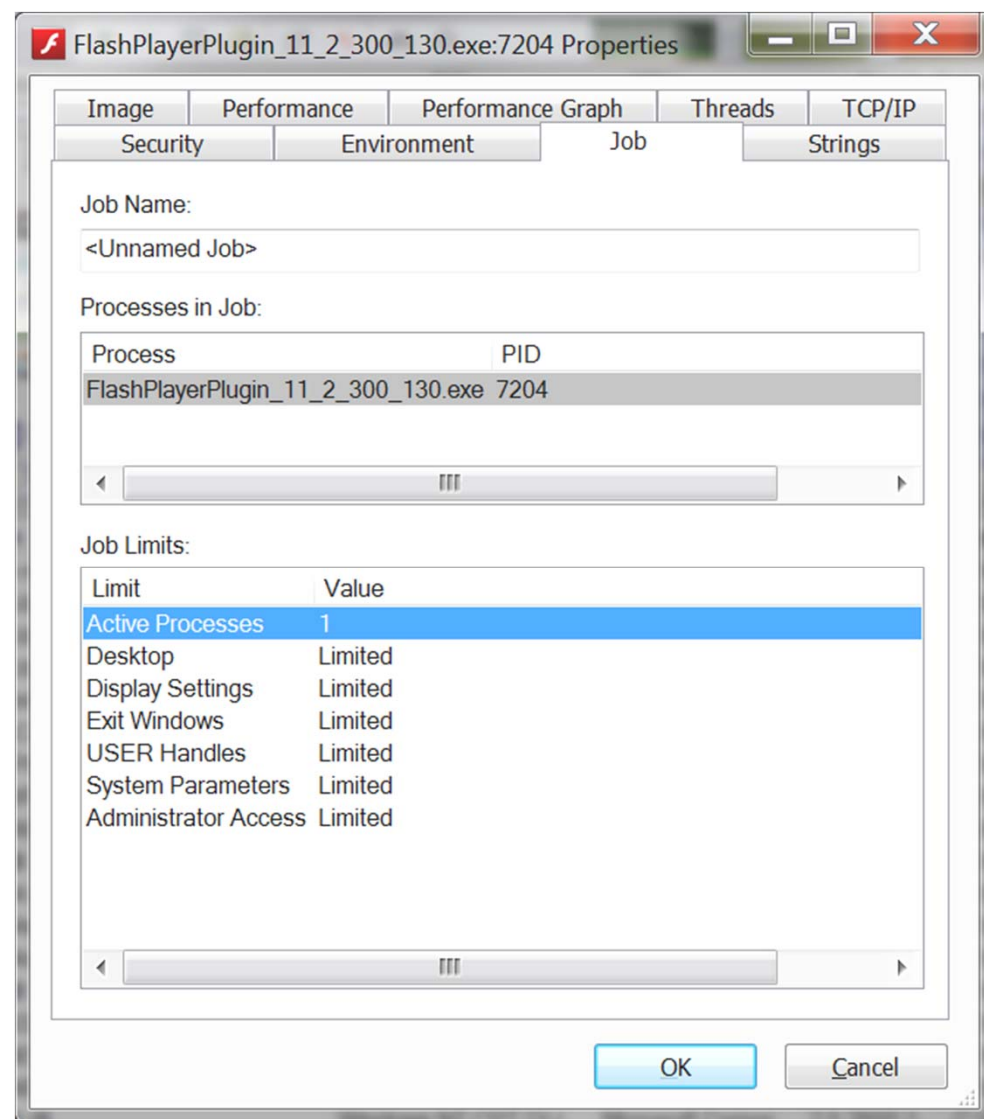
# Sandboxing Flash Player

## Browser:

- Chrome Pepper sandbox
- Firefox NPAPI sandbox
- IE 10 Advanced Protected Mode

## Outside the browser:

- Office 2010 or greater
- Reader X or greater

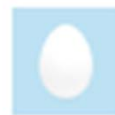


## Other minor improvements

- Safe unlinking in garbage collection
- Random function alignment
- Random NOP insertion
- Constant folding\*

## Other minor improvements

- Safe unlinking in garbage collection
- Random function alignment
- Random NOP insertion
- Constant folding\*



@HaifeiLi

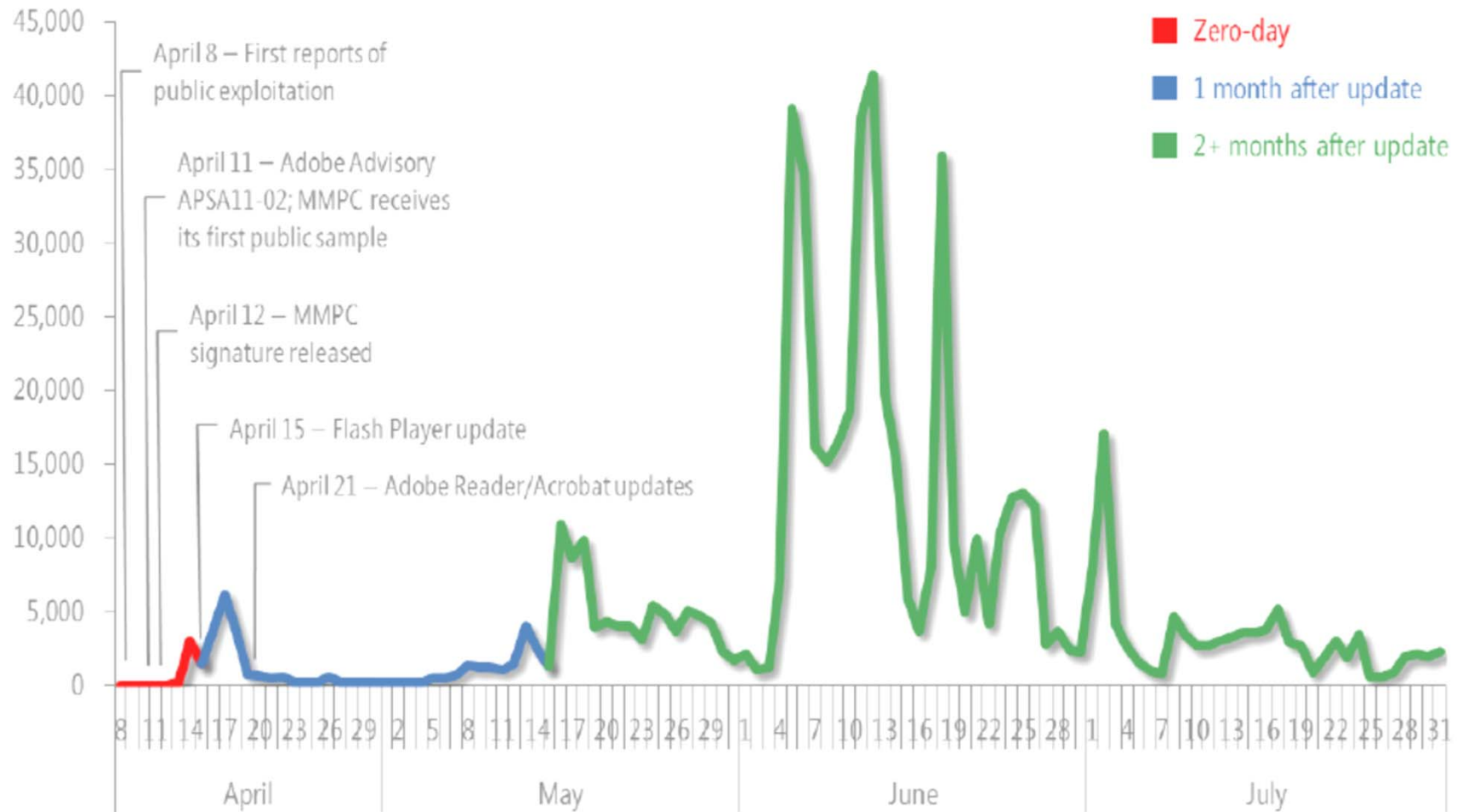
Haifei Li(cocoruder)

Well, this Flash JIT mitigation update clearly killed a concept I had a few days ago that I haven't got chance to test :( Congrats to Adobe!

10 Nov via web

# Updating end-users

Figure 6. Detections of exploits targeting CVE-2011-0611, April–July, 2011



## Update Goals

- Work with AV and IDS vendors to create accurate signatures that will protect end-users until they get the patch (MAPP)
- Reduce the time to update the majority of end-users to minimize the window of opportunity for the exploit



# Updaters

- Flash Player background updater for Windows and Mac
- Chrome updater
- Windows Update for IE 10 & Windows 8
- SCUP & SMS support

# Handling response

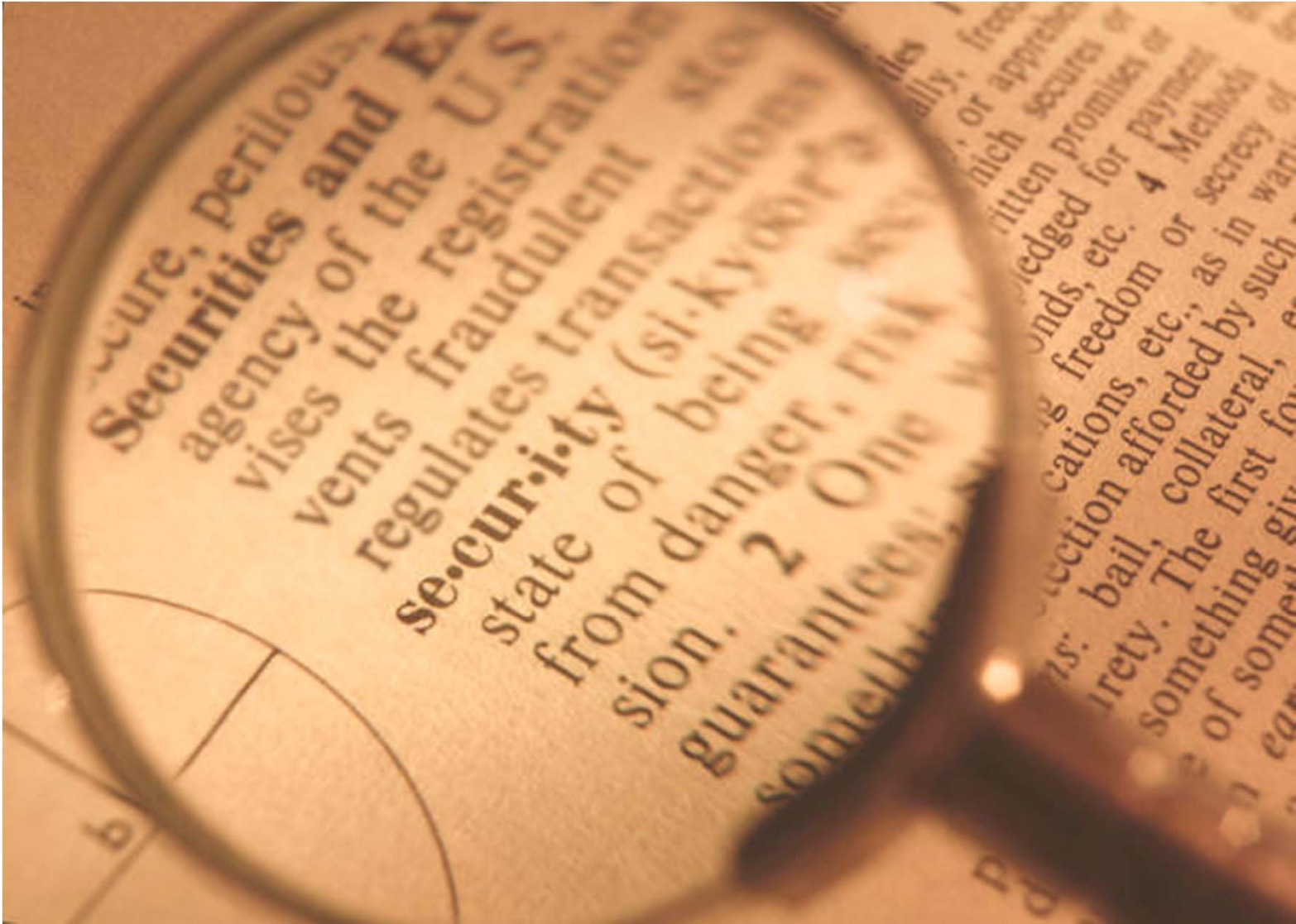


## Incident response strategies

- Be prepared to triage duplicates, triplicates, quadruplicates, etc....
- Set a response timeline goal
- Have a regular update schedule
- Be willing to shift launch dates
- And have tools....

- Open-source AIR application
- View SWF tags, disassembly and binary
- Test AMF services and check for XSS
- Inspect LSOs and settings files
- Execute the SWF in various contexts

# Summary



- Understand your threats
- Advanced, holistic security features are needed to ward off future threats
- Need to utilize both internal and external resources to accomplish goals
- Start early because the best defenses require time to develop

# Always keep moving forward

- Windows 8/IE 10 integration
- Reader XI sandbox improvements



# References

- Security portal: (customer & channel partners) <http://adobe.com/security>
- Advisories and updates: <http://www.adobe.com/support/security/>
- ASSET blog: <http://blogs.adobe.com/asset>
- PSIRT blog: <http://blogs.adobe.com/psirt>
- Documentation Wiki: <http://learn.adobe.com/wiki/display/security/Home>
- Adobe Security on Twitter: [@AdobeSecurity](https://twitter.com/AdobeSecurity)
- Peleus Uhley on Twitter: [@PeleusUhley](https://twitter.com/PeleusUhley)





**Adobe**