How to blackbox test almost anything

Aviram Jenik, CEO



Recipe for finding unknown security holes





Recipe for finding unknown security holes

On every platform



Recipe for finding unknown security holes

On every platform For every programming lang.





Recipe for finding unknown security holes

On every platform For every programming lang. Any product type









Blackbox Testing



Is blackbox testing really that powerful?

This week's major security holes on SecuriTeam.com:

- F5 Denial of Service
- Mozilla Firefox disclosure vulnerability
- **Cisco TCP connection DoS**
- Adobe Flash Player code execution

Datastor Kernel Denial Of Service Vulnerability | 0 Comments

The datastor kernel module in F5 BIG-IP Analytics, APM, ASM, Link Controller, and LTM 11.1.0 before 12.0.0, BIG-IP AAM 11.4.0 before 12.0.0, BIG-IP AFM, PEM 11.3.0 before 12.0.0, BIG-IP Edge Gateway, WebAccelerator, and WOM 11.1.0 through 11.3.0, BIG-IP GTM 11.1.0 through 11.6.0, BIG-IP PSM 11.1.0 through 11.4.1, BIG-IQ Cloud and Security 4.0.0 through 4.5.0, BIG-IQ Device 4.2.0 through 4.5.0, BIG-IQ ADC 4.5.0, and Enterprise Manager 3.0.0 through 3.1.1 allows remote authenticated users to cause a denial of service or gain privileges by leveraging permission to upload and execute code... More >>>

Mozilla Firefox URL Information Disclouser Vulnerabilities | 0 Comments

The Search feature in Mozilla Firefox before 42.0 on Android through 4.4 supports search-engine URL registration through an intent and can access this URL in a privileged context in conjunction with the crash reporter, which allows attackers to read log files and visit file: URLs of HTML documents via a crafted application.. More >>>

Cisco Virtual Topology System TCP Connection Functionality Denial Of Service Vulnerability | 0 comments

Cisco Virtual Topology System (VTS) 2.0(0) and 2.0(1) allows remote attackers to cause a denial of service (CPU and memory consumption, and TCP port outage) via a flood of crafted TCP packets. More ...

Adobe Flash Player And AIR Arbitrary Code Execution Vulnerabilities

Use-after-free vulnerability in Adobe Flash Player before 18.0.0.268 and 19.x and 20.x before 20.0.0.228 on Windows and OS X and before 11.2.202.554 on Linux, Adobe AIR before 20.0.0.204, Adobe AIR SDK before 20.0.0.204, and Adobe AIR SDK & Compiler before 20.0.0.204 allows attackers to execute arbitrary code. More >>>



15 Feb. 2016

16 Feb. 2016

14 Feb. 2016

13 Feb. 2016

What do they have in common?

- 4 different products
- Different attack vectors
- All critical vulnerabilities
- All require patch
- Unpatched systems will be extremely vulnerable
- All could have been discovered during development
- None requires special expertise to exploit (hence, relatively straightforward to discover)
- (probably) found via fuzzing e.g. blackbox testing



Blackbox testing in the real world

This was my week:

Monday	Tuesday	Wednesday
Bluetooth attack (unpaired) on a car hands-free system	Malformed WAV file vulnerability on a car entertainment system	CAN BUS (OBD-II) request caused full lock-up of the car (required towing)





Something about me





Finding those vulnerabilities automatically (machinestyle)





About Beyond Security

We specialize in vulnerabilities





About Beyond Security

We specialize in vulnerabilities and develop tools to find them



Our Technology







AVDS

beSTORM

SecuriTeam.com



Our Technology - AVDS



AVDS

Automated Vulnerability Detection System ("VA")

Know that your network is **Safe**

Latest, runats	(0 - 10 of 63)	next+ end+	
Vulnerability Name	State	Priority	
-NetBIOS Information Retrieval	Open	Critical	
-ICMP Timestamp Request	Open	Moderate	
- <u>Telnet Detection</u>	Open	Moderate	
- <u>404 check</u>	Open	Moderate	
- <u>404 check</u>	Open	Moderate	
-Passwordless Alacatel ADSL Modem	Open	Moderate	
-PPTP Detection	Open	Moderate	
-MySQL Server Version Detection	Open	Moderate	
- <u>Directory Scanner</u>	Open	Moderate	100
- <u>Directory Scanner</u>	Open	Moderate	
-Directory Scanner	Open	Moderate	
-Directory Scanner	Open	Moderate	and the second
-MYSOL Server Version Detection	Open	Moderate	
-PPTP Detection	Open	Moderate	
«Passwordiess Alacatel ADSL Modem	Open	Moderate	
- <u>404 ched</u> ;		Moderate	
CMP Timestamp Request	Open		
NetBIOS Information Retrieva		Contract of the local division of the local	-
Vuinerability			
Training and the second s		2.72	
NORMON NO.		Shannar	R 1000 10.00 10.00
	10.000		The second second
		the second secon	-
		-	
	0		TODA
AVD	5	pes	TORM

SecuriTeam.com

2 a curity

Known Vulnerabilities



AVDS

(Automated Vulnerability Detection System)

- Everything that talks 'IP".
- Agent-less, providing a real 'hacker' view
- Scalable from 64 to many hundreds of thousands of systems
- Very powerful management and automation tools





Our Vision with AVDS

- Scanning 1-4/year is just not enough any more
- Vulnerability Management is an active process
- scans must be done on a regular basis:
 - > New vulnerabilities are discovered every day
 - > The network is dynamic (new ports, services, hosts)
- Needs to be a dedicated, robust platform (similar to Firewall/Proxy/IPS)

The objective of Vulnerability Management is to KNOW, at any given time, what the risks are in your infrastructure so that they can be managed.



Our Technology - SecuriTeam



6

10

SecuriTeam.com

Security portal / Knowledge Source



the set of the set	Tel: 10 (_
Contraction of the local division of the loc		-	
the state of the			
and interior			
Terrestant, while			
and the second			
BORD BRANCH		1000	-
Thursday	-		
_			
			And in case of
And in case of the local division of the loc		the state of the	100
the second second			
Contraction of the local division of the loc			



SecuriTeam.com

AVDS

beSTORM

Information

			Search	
Securi7			All Sections	Search
ocourri	Security News - Security Reviews - Exploits - Tools - UNIX Focus - Wi Bloos News	ndows Focus -		
e d	Featured Articles:		Blogs	24 Dec. 200
Securiteam" Securiteam" Securiteam	Adobe Flash Player ActiveX Control Universal Cross-Site Scripting Yulnerability The specific flaw exists in the Flash Player ActiveX Control's handling navigateFoURL APL, which takes two arguments, a URL and the nam to be navigated. More ••• Apple Mac OS X Software Update Command Execution Yulnerability Authorability within Annio Mar OS X's Software Undate mechanism	20 Dec. 2007 of the e of the frame 18 Dec. 2007	 From descri Tis the sea Cryptome: I real-time at Hushmail se New Securi Solutions Orkut virus/ 	ption to exploit son VSA has cess to rivers ly Threats & worm on the
Home Ask the Team Mailing Lists	attackers to cause it to execute arbitrary code by creating a malicious server. More …	is update	loose	More >>
Advertising Info Advisories	Vulnerability in SMBv2 Allows Code Execution (MS07-063)	12 Dec. 2007	Exploits	23 Dec. 2003
Blogs	A remote code execution vulnerability exists in the SMBv2 protocol to allow a remote anonymous attacker to run code with the privileges i	of the	Socket Con Can Reveal	nection Timing Information
SecuriTeam** in Your Inbox	logged-on user. More ···		About Netw Configuration	ork an (Exploit)
	arrisa	25 Dec. 2007	Microsoft W	indows Jacuino Sarvica
vew vulnerability? vew tool? Fell us	Zoom Player Unicode Buffer Oxerflow ImgSvr Directory Traversal		Stad: Overf Vulnerabilit Exploit)	iow / (MS07-065,
Our PGP key).	VideoLAN VLC Buffer Overflow and Format String email Applan Enterprise Business Suite DoS	20 Dec. 2007	Clamav-milt Sendmail Al Command E	er and low Arbitrary xecution
Black Hat	Apple Mac OS X mount_smbfs Stad: Based Buffer Overflow Vulnerab umm Appleinton Inspection Vulnerability in Cisco Firewall Services Module	olity	(Emploit) Apple Mac 0 Vulnerabiliti (mount smi	IS X SMB es ofs and
DC 2000	Adobe Flash Player JPG Processing Heap Overflow Vulnerability ClamAV libdamav MEW PE File Integer Overflow Vulnerability		smbutil) Ø OpenSSL St	iLv2 Client
	imesh IMWebControl Class Heap Overflow St. Bernard Open File Manager Heap Overflow Vulnerability	18 Dec. 2007	SurgeMail V Header Dos	(ebmail Host
	Mean Investoritorio class Heap Orentow St. Bernard Open File Manager Heap Orentow Vulnerability Tpend Micro SeptestBrotect StBocSor 4II Toreo ne Metbod Export tea (v	idoecabilita	Header Dos	id mailsist()"
Keeleen	Application Inspection Viruneiraaliity in cisco irie wali Services Acquis Adobe Flash Player JPG Processing Heap Overflow Vulnerability ClamAV libdamav MEW PE File Integer Overflow Vulnerability mm	18 Dec. 2007	(moon Sm smbubl) OpenSSL St Crash (NUL	urs and suv2 Client L Reference) folgeneit Hoef
Black Hat	roza Apple Mac OS X mount_smbfs Stadi. Based Buffer Overflow Vulnerat amm		(Explort)	
N22				

SecuriTeam.com

Security portal / information source

- All information/mailing lists are free
- Global gathering place for IT
 Security Professionals and Hackers
- One of the leading portals worldwide on vulnerabilities and exploits

Securit			Taba
4.15	1111 (11 4)		
<u>×</u>			
100	and south the real of	10.00	
100		-	1.1.1.1
1000	Constraint State		
The state			
100	A CONCRETE OFFICE	- 1	
Frank .		1-1-1-1	
0	1.35 Martine	F	



AVDS



beSTORM



Among our customers



How are cyber attacks done?

"The reality is that the most important issues are the vulnerabilities and the techniques used to exploit them, not the country that appears to be the source of the attack"

- Gartner

How was the recent cyber-incident done? Most likely by a vulnerability that is easy to uncover and patch



The theory behind Blackbox testing



The most secure system







Is completely useless





Inputs are the problem



For the programmer



Inputs are the problem

3 1/5

For the attacker



Turing machine





memory tape

The basic software model



What is blackbox testing?

- Testing by attacking the inputs and observing output/behavior
- Does not use the source code
- Does not assume knowledge about the system



Doesn't QA solve this?

QA: Testing if a good input => good result

Fuzzing: Testing if a malformed input => good result for the attacker!



What is blackbox testing?

- Testing by attacking the inputs and observing output/behavior
- Does not use the source code
- Does not assume knowledge about the system

The system is a black box



What is blackbox testing?

- Testing by attacking the inputs and observing output/behavior
- Does not use the source code
- Does not assume knowledge about the system

The system is a black box

This is how almost all security holes are discovered today



The theory: Generate all possible combinations







000...000 000...001 000...010

111...111

All possible inputs ==> All possible outputs





All possible outputs ==> All possible security vulnerabilities will be triggered



The drawback: 1KB request = 2¹⁰⁰⁰=10³⁰⁰ combinations





MS



Blackbox Testing French people



Example

Blackbox Testing French people

Tous les verbes de la langue française Les tableaux modèles

Les règles de conjugaison Les règles d'accord

LAROUSSE CHAMBER:

DICTIONNAIRE FRANÇAIS-ANGLAIS ENGLISH-FRENCH

NOUVELLE ÉDITION



DES PLANCHES INÉDIT THÉMATIQUES ILLUSTRÉES







2° ÉDITION avec 440 exercices

Maïa Grégoire



Step 1: Generate all possible valid protocol requests (by crawling through the BNF description of the protocol)



Step 1: Generate all possible valid protocol requests (by crawling through the BNF description of the protocol)

==> Guaranteed to cover the entire protocol



Step 1: Generate all possible valid protocol requests (by crawling through the BNF description of the protocol)

==> Guaranteed to cover the entire protocol

Step 2: "Fuzz" (=attack) each field in each combination



Step 1: Generate all possible valid protocol requests (by crawling through the BNF description of the protocol)

==> Guaranteed to cover the entire protocol

Step 2: "Fuzz" (=attack) each field in each combination

Attacks:

- Buffer overflow: AAAA...AA
- Format string: %n
- Null character: 0x00
- XML attacks: < and >
- Space
- Various encodings



Problem: Buffer overflow testing requires many combinations:



Problem: Buffer overflow testing requires many combinations:

USER AAAAAAA...A (n times)

What is `n'?



Problem: Buffer overflow testing requires many combinations:

USER AAAAAAA...A (n times)

What is `n'?

Naive Solution: Test everything.



Problem: Buffer overflow testing requires many combinations:

USER AAAAAAA...A (n times)

What is `n'?

Naive Solution: Test everything.

USER A

USER AA

...

USER AAA



Problem: Buffer overflow testing requires many combinations:

USER AAAAAAA...A (n times)

What is `n'?

Naive Solution: Test everything.

USER A

USER AA

. . .

USER AAA

"n" combinations



Problem: Buffer overflow testing requires many combinations:

USER AAAAAAA...A (n times)

What is `n'?

Smarter option: Scale quickly 2ⁿ

USER A (2º)

USER AA (2^1)

USER AAAA (2²)

USER AAAAAAAA (2³)

 $\log_2(n)$ combinations



Smarter solution can test buffers from 1 byte to 64KB in <u>16</u> steps (naïve method takes 65,535 steps)

Still covers small buffer sizes (2, 4, 8) Covers medium buffer sizes (1024, 2048) And covers large buffers (32,000 and 64,000)



Smart Fuzzing

Conclusion: It's possible to do an 'exhaustive' testing while taking a few shortcuts to reduce the combination count without reducing quality



How to blackbox almost everything

Step 1: map all your inputs – in production

- File inputs
- Network
 - IP
 - Wireless?
 - RFID?
- Library calls
- Command line parameters



Why file input can be especially dangerous

"preview" - ANI attack





Why file input can be especially dangerous

"preview" - ANI attack





Who determines risk?

Not you!



Who determines risk?

Attackers attack what's easy and not where you ask them





security

How to blackbox almost everything

Step 2: determine your "protocols"



Protocols

- Network: your RFC (or spec-based) protocol
- File: accepted file formats
- Library: DLL/ActiveX Interface



How to blackbox almost everything

Step 3: Start testing



Ingredients

- **1. Test Module description**
- 2. Generator
- 3. Monitor



8

Test Module

- Something that can describe "many" "different" sessions (=attacks)
- Protocol coverage is key



Example: beSTORM BSP file format

```
<SC Name="ICAP Request">
<SE Name="ICAP Method">
<S Name="ICAP Method Enumerating">
 <E Name="ICAP Methods">
 <C Name="REQMOD Method" ASCIIValue="REQMOD" />
 <C Name="OPTIONS Method" ASCIIValue="OPTIONS" />
 </E>
<S Name="ICAP Method Overflow">
 <B Name="ICAP Method Overflowing" ASCIIValue="RESPMOD" />
</SE>
<S Name="Request Line">
<C Name="Space" ASCIIValue=" " />
<B Name="icap Prefix" ASCIIValue="icap" />
<C Name="ColonSlashes" ASCIIValue="://" />
<B Name="Address" ASCIIValue="10.50.10.71" />
</SC>
```



Example: beSTORM BSP for file fuzzing

```
<M Name="TGA" >
      <P Name="TGA Protocol" >
         <SP Name="Writer" Library="File Utils.dll"
  Procedure="Write">
           <S Name="Path" > <VB Name="Whatever" Description="Path to store
  files" NoDefaultTypes="1" ASCIIValue="c:\\temp" /> 
           <S Name="Directory Splitter" >
             <VB Name="Whatever" Description="Directory Splitter
  size" NoDefaultTypes="1" ASCIIValue="2" />
           <S Name="Extension" >
             <VB Name="Whatever" Description="Extension"
  NoDefaultTypes="1" ASCIIValue="tga" />
           <SC Name="Data" >
             <S Name="Color-mapped images" >
               <L Name="Identsize" ConditionedName="Image
  Identification Field" Size="1" />
               <B Name="Colour Map Type" Default="0x00"
  MaxBytes="1" />
               <B Name="Image Type Code" Default="0x02"
  MaxBytes="1" />
               <B Name="Color Map Origin" Default="0x00,0x00"
```

Generator

- Something that can take the module description and send it to the program:
 - Over the network
 - By creating a file
 - By invoking a DLL function



Monitor

- Possibly the most important component
- So you're generating millions of attacks: but how do you know you succeeded?



Monitoring

Monitor for:

- Memory exceptions ("first chance exceptions")
- Program stops responding
- Errors in Logs (via regex)
- Connect the monitor with the generator to correlate





Easy to use and extend

- Windbg
- gdb



Key factors

- Automation
- Re-creating the attacks
- Ensuring protocol coverage (not code coverage!)



Report Sample



Thank you!

Questions?

aviram@beyondsecurity.com

