

Inspector Gadget

*Automated Extraction of Proprietary
Gadgets from Malware Binaries*

Clemens KOLBITSCH Thorsten HOLZ
Engin KIRDA Christopher KRUEGEL

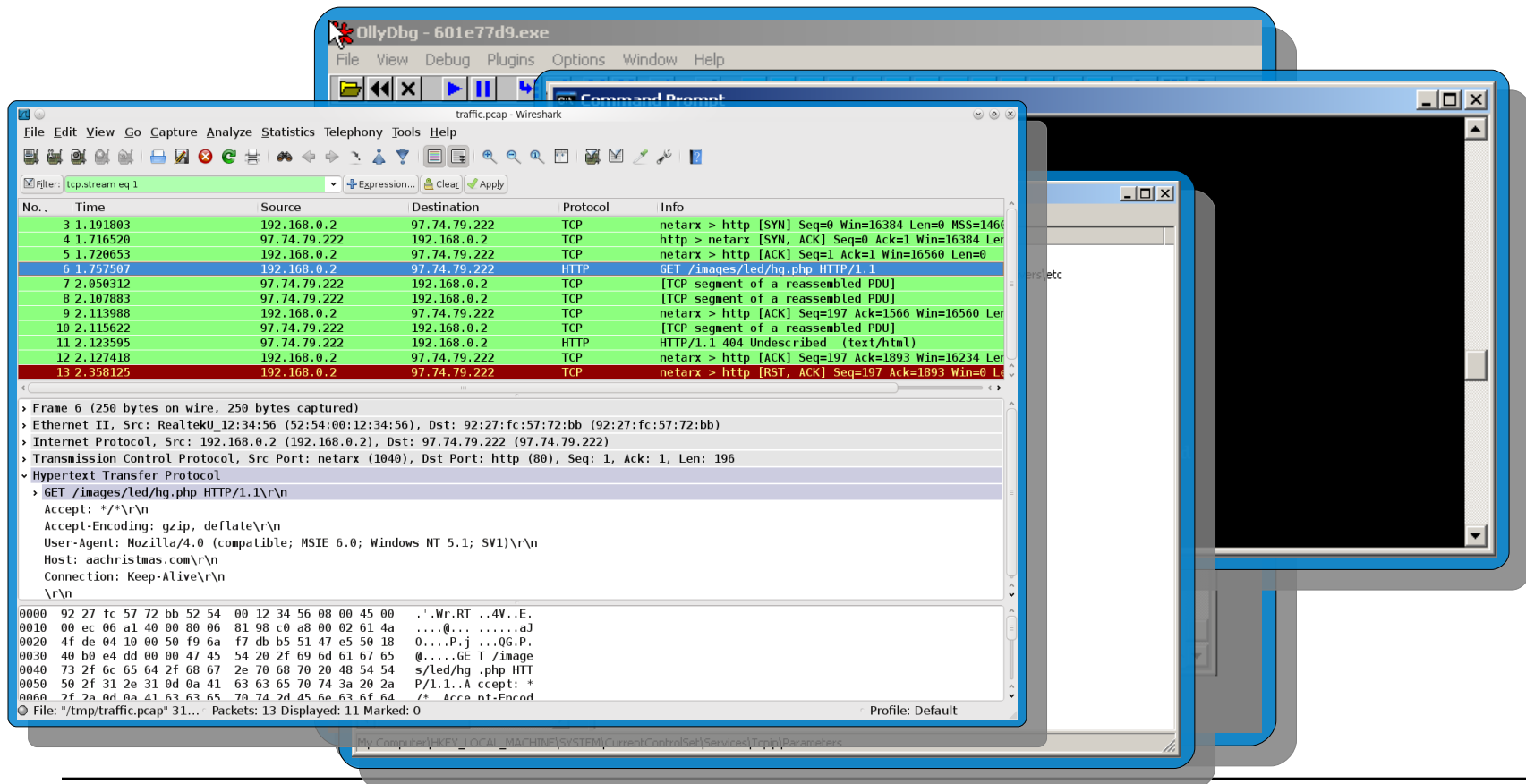
ck@iseclab.org

Int. Secure Systems Lab
Vienna University of Technology, Institute Eurecom Sophia Antipolis, UC Santa Barbara

Motivation

Int. Secure Systems Lab
Vienna University of Technology

- Analysis of malicious code is challenging



Motivation

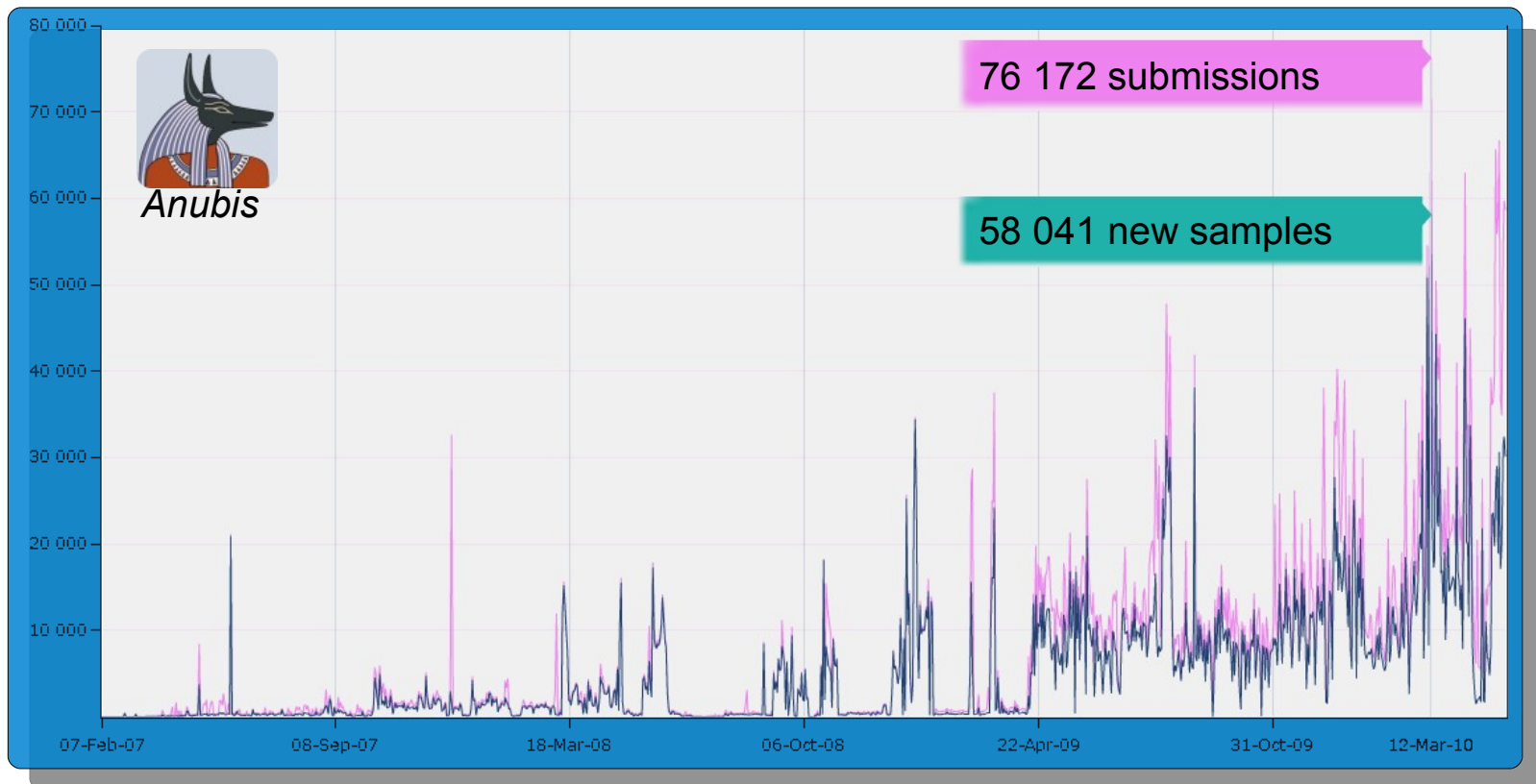
Int. Secure Systems Lab
Vienna University of Technology

- Analysis of malicious code is challenging
- Looking at the inner workings of every samples has become infeasible
 - ... due to various *obfuscation* techniques
 - ... due to *analysis resistance* (e.g., *anti-debugging techniques*)
 - ... due to the huge *number* of malware *families / variants*

Motivation

*Int. Secure Systems Lab
Vienna University of Technology*

- Analysis of malicious code is challenging



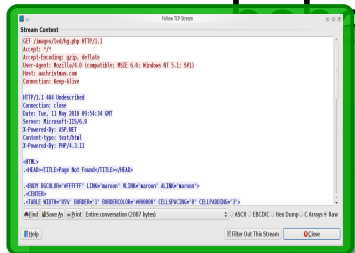
Motivation

Int. Secure Systems Lab
Vienna University of Technology

- Analysis of malicious code is challenging
- Looking at the inner workings of every samples has become infeasible
 - ... due to various *obfuscation* techniques
 - ... due to *analysis resistance* (e.g., *anti-debugging techniques*)
 - ... due to the huge *number* of malware *families / variants*
- Results of dynamic analysis is cluttered by other behavior sample is capable of

Motivation

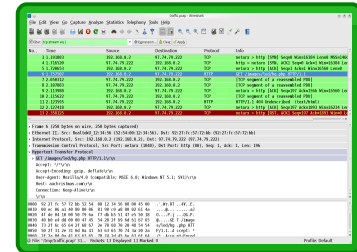
- Results of dynamic analysis is cluttered by other behavior sample is capable of



binary update

armzasn.net
kevflnwroo.com
dzqbpieiy.info
rqkixea.biz
komug.net
vhiax.org

C & C location



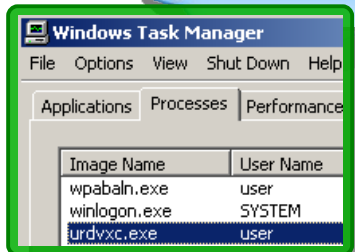
C & C communication



target selection

spam templating

220 mx.google.com
250-google
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-STARTTLS



Motivation

Int. Secure Systems Lab
Vienna University of Technology

- Results of dynamic analysis cluttered by other behavior sample is capable of
- Dynamic analysis is very resource consuming...
- ... and only provides temporary snapshot
 - malicious behavior might dependent on
 - analysis date & time
 - analysis environment (e.g., username, host OS, ...)
 - availability of remote resources (e.g., C&C hosts)
 - needs to be repeatedly performed on single sample
 - at different points in time
 - preferably on different systems
 - even more time/resource consuming

Motivation – Inspector

Int. Secure Systems Lab
Vienna University of Technology

Wouldn't it be cool if we were able to extract a single behavior into a standalone component and use this to re-invoke the behavior?

- Removes clutter from analysis results
- Independent of other malicious activity
 - can be executed without virtual environment
- Easy to replay in a different situation such as
 - point in time
 - operating system

Motivating Example

Int. Secure Systems Lab
Vienna University of Technology

- *Conficker* Domain Generation Algorithm (DGA)
 - decides which remote host to contact for C&C
 - domain depends on *current time*
 - current time is fetched from a remote host (e.g., msn)

Motivating Example

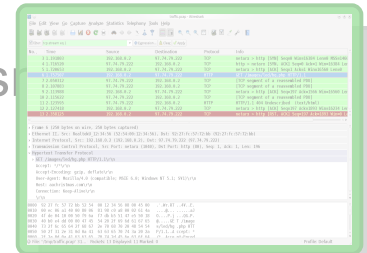
- Conficker Domain Generation Algorithm (DGA)

decides which re... ac... C&C
depends...
C & C
location

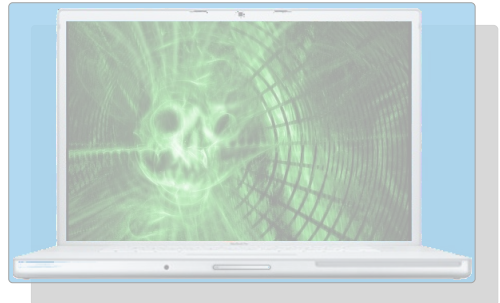
```
armzasn.net  
kevflnwroo.com  
dzqbpieiy.info  
rqkixea.biz  
komug.net  
vhiax.org
```

binary
update

C & C
location



C & C
communication

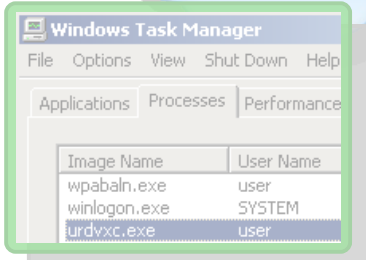
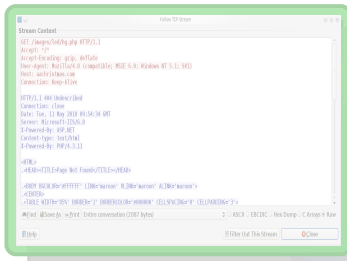


component
installation

target
selection

spam
templating

```
220 mx.google.com  
250-google  
250-PIPELINING  
250-SIZE 10240000  
250-VRFY  
250-STARTTLS
```



Outline

Int. Secure Systems Lab
Vienna University of Technology

- Motivation
 - dynamic analysis reveals limited, temporary behavior
- Behavior analysis & extraction
 - storing identified behavior into *gadget*
- Behavior re-invocation
 - gadget player
 - gadget inversion
- So... again, why...?
 - benefit recap
- Gadget examples

Behavior Analysis and Extraction

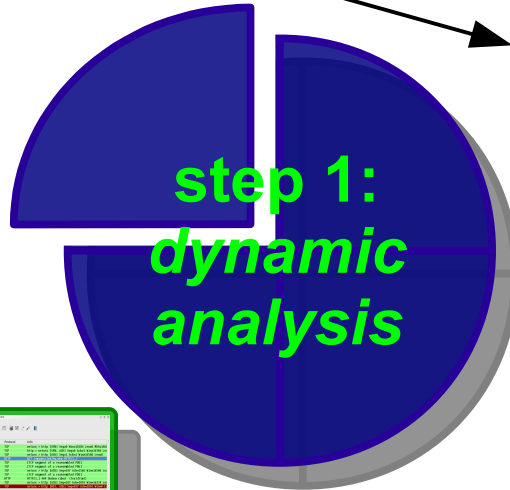
Extraction Overview

*Int. Secure Systems Lab
Vienna University of Technology*



Extraction Overview

Int. Secure Systems Lab
Vienna University of Technology



- control flow & instructions
- API taint dependencies
- memory accesses

Analysis Report for setup.exe - submitted on 05/13/10, 19:06:44 UTC

Run-into Dlls

Module Name	Base Address	Size
C:\WINDOWS\system32\CLBCATQ.DLL	0x76FD0000	0x0007F000
C:\WINDOWS\system32\COMRes.dll	0x77505000	0x000C5000
C:\WINDOWS\system32\WINNLS.dll	0x771B0000	0x000A4000
C:\WINDOWS\system32\SETUPAPI.dll	0x77320000	0x000F3000
C:\WINDOWS\system32\CRYPTSP.dll	0x775A0000	0x0005F000
C:\WINDOWS\system32\MSASN1.dll	0x77520000	0x00012000
C:\WINDOWS\system32\apphelp.dll	0x77B40000	0x00022000
C:\WINDOWS\system32\urlmon.dll	0x7E1E0000	0x000A2000
C:\WINDOWS\system32\shdocvw.dll	0x7E290000	0x00171000

2.a) setup.exe - Registry Activities

Key	Name	New Value
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell	Common AppData	C:\Documents and Settings\All Users\Desktop
	Application Data	C:\Documents and Settings\All Users\Desktop
	Common Documents	C:\Documents and Settings\All Users\Documents
	Common Documents	C:\Documents and Settings\All Users\Documents
	Common Start Menu	C:\Documents and Settings\All Users\Start Menu

Windows Task Manager

Image Name	User Name
wpabnl.exe	user
winlogon.exe	SYSTEM
lurdxvc.exe	user

Find interesting behavior that is to be extracted.
Example: *Hm, to which domain am I connecting here??*

Extraction Overview

Int. Secure Systems Lab
Vienna University of Technology



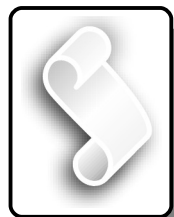
Analysis Report for setup.exe - submitted on 05/13/10, 19:06:44 UTC

Runtime DLLs	Module Name	Base Address	Size
	C:\WINDOWS\system32\CLBCATQ.DLL	0x78FD0000	0x0007F000
	C:\WINDOWS\system32\CCMRes.dll	0x77050000	0x000C0000
	C:\WINDOWS\system32\WINEVT.dll	0x771B0000	0x000A0000
	C:\WINDOWS\system32\SETUPAPI.dll	0x77200000	0x000F0000
	C:\WINDOWS\system32\CRYPT32.dll	0x77A00000	0x00090000
	C:\WINDOWS\system32\MSASN1.dll	0x77BD0000	0x00012000
	C:\WINDOWS\system32\apphelp.dll	0x77840000	0x00022000
	C:\WINDOWS\system32\urlmon.dll	0x7E1E0000	0x000A2000
	C:\WINDOWS\system32\shdocvw.dll	0x7E290000	0x00171000

Static Virus Scanner
Trojan.Win32.StartPage (Sig:Id1388957)

Z.a) setup.exe - Registry Activities

Registry Values Modified:	Key	Name	New Value
	HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders	Common AppData	C:\Documents and Settings\All Users\ Application Data
	HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders	Common Desktop	C:\Documents and Settings\All Users\ Desktop
	HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders	Common Documents	C:\Documents and Settings\All Users\ Documents
	HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders	Common Start Menu	C:\Documents and Settings\All Users\Start Menu



control flow &
instructions

Map selected behavior to analyzed
process & thread, API accesses and
control flow.

outcome: API call / flow position

Extraction Overview

Int. Secure Systems Lab
Vienna University of Technology



Analysis Report for setup.exe - submitted on 05/13/10, 19:06:44 UTC

Runtime Dlls

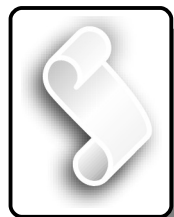
Module Name	Base Address	Size
C:\WINDOWS\system32\CLBCATQ.DLL	0x78FD0000	0x0007F000
C:\WINDOWS\system32\CCMRes.dll	0x77050000	0x000C0000
C:\WINDOWS\system32\WININET.dll	0x771B0000	0x000A4000
C:\WINDOWS\system32\SETUPAPI.dll	0x77320000	0x000F3000
C:\WINDOWS\system32\CRYPT32.dll	0x77A80000	0x00096000
C:\WINDOWS\system32\MSASN1.dll	0x77B20000	0x00012000
C:\WINDOWS\system32\apphelp.dll	0x77940000	0x00022000
C:\WINDOWS\system32\urlmon.dll	0x7E1E0000	0x000A2000
C:\WINDOWS\system32\shdocvw.dll	0x7E290000	0x00171000

Static Virus Scanner
Trojan.Win32.StartPage (Sig:141388957)

2.a) setup.exe - Registry Activities

Registry Values Modified:

Key	Name	Value
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders		Common
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders		Common
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders		Common
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders		Common

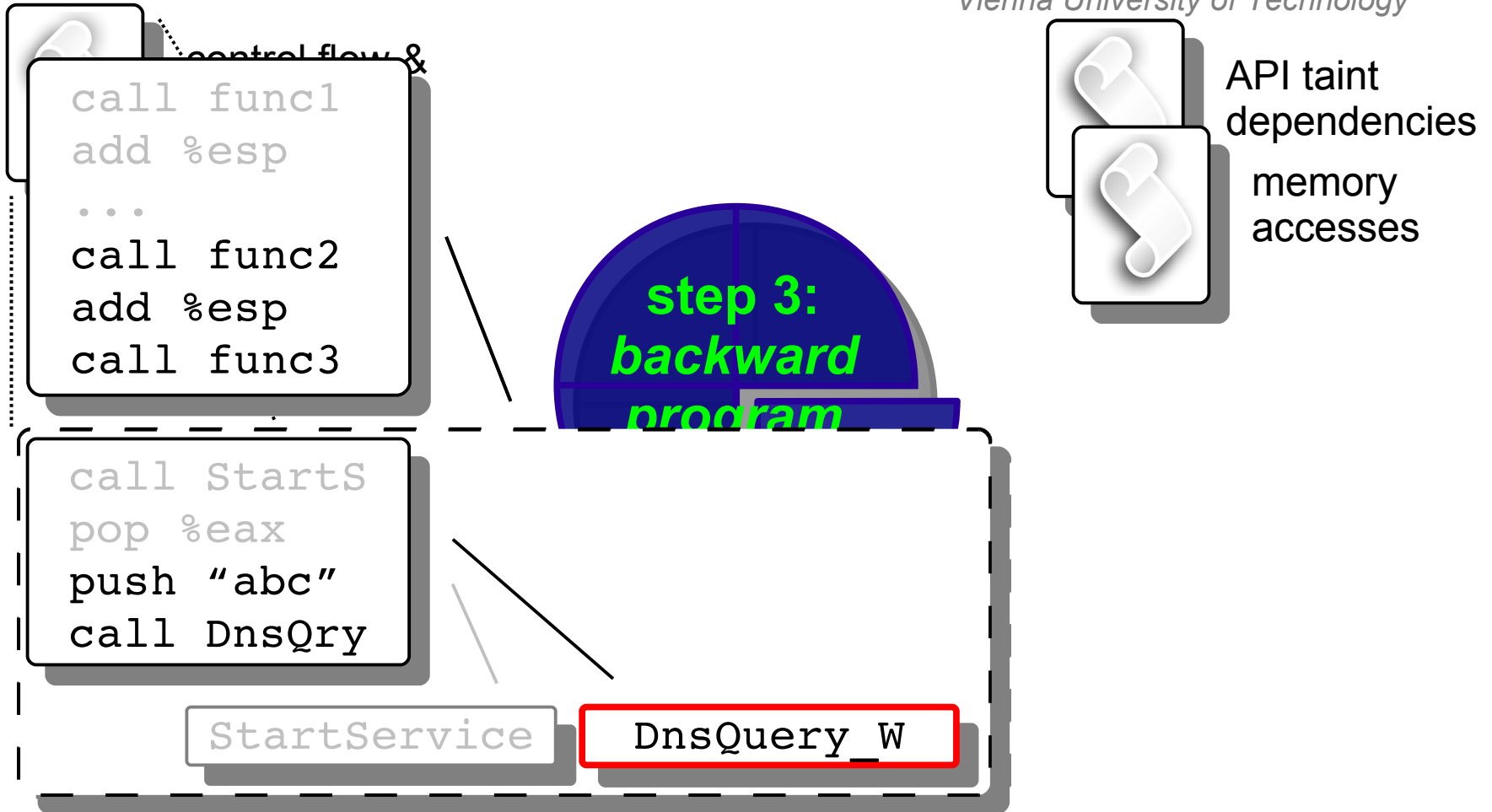


control flow & instructions

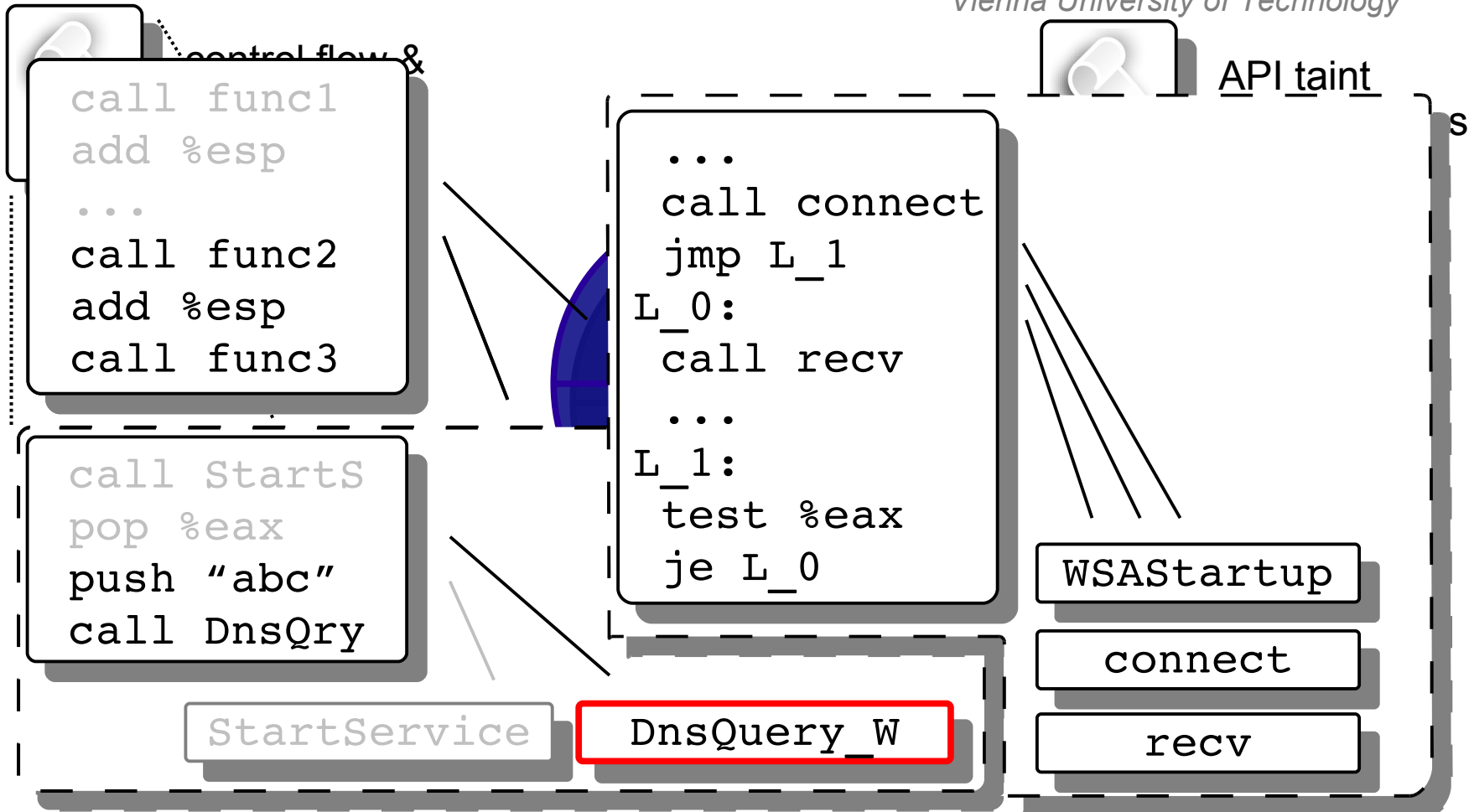
Find and suggest data manipulating instructions after chosen API call. Possibly refine chosen position to include the data processing.

Extraction Overview

Int. Secure Systems Lab
Vienna University of Technology

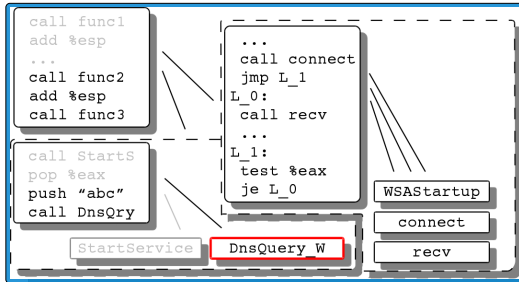


Extraction Overview



Extraction Overview

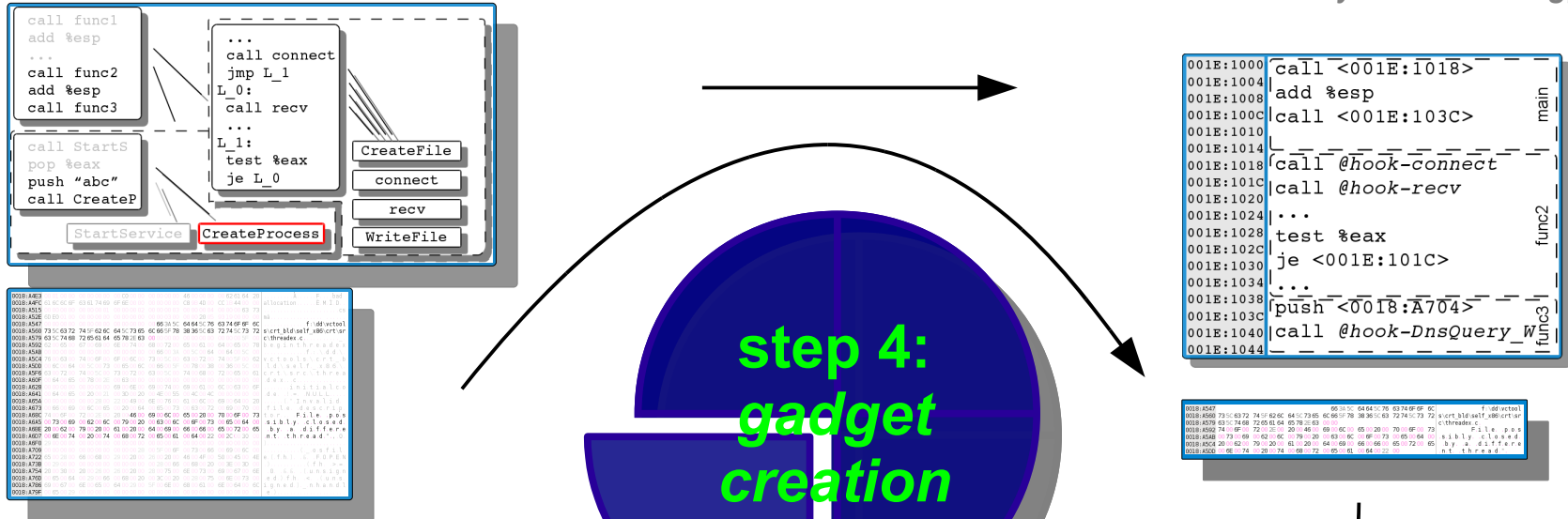
Int. Secure Systems Lab
Vienna University of Technology



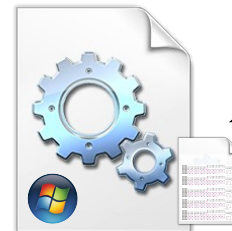
```
0018:A4E3 00 01 00 00 00 00 00 00 00 C0 00 00 00 00 00 00 00 46 00 00 00 00 62 61 64 20 .....Ã....F...bad
0018:A4FC 61 6C 6C 6F 63 61 74 69 6F 6E 00 00 00 00 00 00 00 C8 98 4D 00 CC 1B 44 00 00 allocation.....ËMID.
0018:A515 00 00 00 00 00 00 00 01 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 63 73 .....
0018:A52E 6D E0 01 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 20 05 93 19 00 00 00 mã.....
0018:A547 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 3A 5C 64 64 5C 76 63 74 6F 6F 6C f:\add\vctool
0018:A560 73 5C 63 72 74 5F 62 6C 64 5C 73 65 6C 66 5F 78 38 36 5C 63 72 74 5C 73 72 s\crt_bld\self_x86\crt\sr
0018:A579 63 5C 74 68 72 65 61 64 65 78 2E 63 00 00 00 00 00 00 00 00 00 00 00 5F 00 c\threadex.c.....
0018:A592 62 00 65 00 67 00 69 00 6E 00 74 00 68 00 72 00 65 00 61 00 64 00 65 00 78 begin\threadex
0018:A5AB 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 00 3A 00 5C 00 64 00 64 00 5C 00 _f:\add\
0018:A5C4 76 00 63 00 74 00 6F 00 6F 00 6C 00 73 00 5C 00 63 00 72 00 74 00 5F 00 62 v\tools\crt_b
0018:A5DD 00 6C 00 64 00 5C 00 73 00 65 00 6C 00 66 00 5F 00 78 00 38 00 36 00 5C 00 ld\self_x86\
0018:A5F6 63 00 72 00 74 00 5C 00 73 00 72 00 63 00 5C 00 74 00 68 00 72 00 65 00 61 00 61 crt\src\threa
0018:A60F 00 64 00 65 00 78 00 2E 00 63 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 dex.c.....
0018:A628 00 00 00 00 00 00 00 00 69 00 6E 00 69 00 74 00 69 00 61 00 6C 00 63 00 6F .....initialco
0018:A641 00 64 00 65 00 20 00 21 00 3D 00 20 00 4E 00 55 00 4C 00 4C 00 00 00 00 00 del\=..NULL...
0018:A65A 00 00 00 00 00 00 28 00 22 00 49 00 6E 00 76 00 61 00 6C 00 69 00 64 00 20 .....("Invalid.
0018:A673 00 66 00 69 00 6C 00 65 00 20 00 64 00 65 00 73 00 63 00 72 00 69 00 70 00 file\descrip
0018:A68C 74 00 6F 00 72 00 2E 00 20 00 46 00 69 00 6C 00 65 00 20 00 70 00 6F 00 73 ton\file.pos
0018:A6A5 00 73 00 69 00 62 00 6C 00 79 00 20 00 63 00 6C 00 6F 00 73 00 65 00 64 00 sibly\close.d
0018:A6BE 20 00 62 00 79 00 20 00 61 00 20 00 64 00 69 00 66 00 66 00 65 00 72 00 65 .by\adifferen
0018:A6D7 00 6E 00 74 00 20 00 74 00 68 00 72 00 65 00 61 00 64 00 22 00 2C 00 30 00 nt\threadad"...
0018:A6F0 29 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0018:A709 00 00 00 00 00 00 00 00 00 00 28 00 5F 00 6F 00 73 00 66 00 69 00 6C 00 .....(\_osfil
0018:A722 65 00 28 00 66 00 68 00 29 00 20 00 26 00 20 00 46 00 4F 00 50 00 45 00 4E e(f.h.)&..FOPEN
0018:A73B 00 29 00 00 00 00 00 00 00 00 00 00 00 28 00 66 00 68 00 20 00 3E 00 3D 00 ..).....(f.h.)>=
0018:A754 20 00 30 00 20 00 26 00 26 00 20 00 28 00 75 00 6E 00 73 00 69 00 67 00 6E 0 ..&..(unsign
0018:A76D 00 65 00 64 00 29 00 66 00 68 00 20 00 3C 00 20 00 28 00 75 00 6E 00 73 00 ed).f.h.<..(uns
0018:A786 69 00 67 00 6E 00 65 00 64 00 29 00 5F 00 6E 00 68 00 61 00 6E 00 64 00 6C igned)_nhandl
0018:A79F 00 65 00 29 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 e)..
```



Extraction Overview



Extract *gadget* (standalone Dll) that can be imported into any (binary) application offering *environment hooks*.



Gadget Replay

Gadget Player

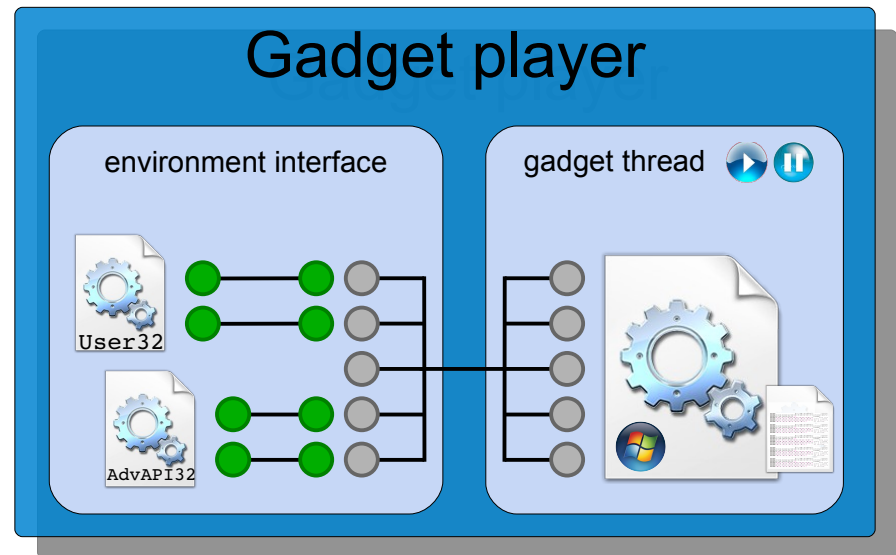
Int. Secure Systems Lab
Vienna University of Technology

- As library, the gadget can be reused in many areas
 - statically linked into the application
 - dynamically loadable
- ... but, application must confine gadget execution
 - handle crashes (e.g., possible, invalid memory accesses)
 - one possibility: code emulation
 - here: separate, monitored thread with signal handling
- ... and mediate accesses to the *host OS*
 - gadgets are guaranteed to contain no calls to system or API functionality directly
 - each access is done through *environment hooks*

Gadget Player

Int. Secure Systems Lab
Vienna University of Technology

- Host OS accesses mediation: *environment hooks*
 - every system / API call is redirected to the gadget player (using a multiplexor function)
 - player has the possibility to *sanitize* and/or *manipulate* call parameters
 - if player decides to allow the API invocation, call and parameters are *forwarded* to the actual implementation (e.g., inside a Windows library)



Gadget Inversion

Int. Secure Systems Lab
Vienna University of Technology

- Player can use gadget as *transformation oracle*
 - *input* is transformed into *output*, depending on algorithm implemented by gadget
 - example: sample reads local data, obfuscates, and transmits to remote host



Gadget Inversion

Int. Secure Systems Lab
Vienna University of Technology

- Player can use gadget as *transformation oracle*
 - *input* is transformed into *output*, depending on algorithm implemented by gadget
 - example: sample reads local data, obfuscates, and transmits to remote host
- In many scenarios, the inverse algorithm would be interesting, however
 - we capture obfuscated traffic and want the plain-text data that has been transmitted



Gadget Inversion

Int. Secure Systems Lab
Vienna University of Technology

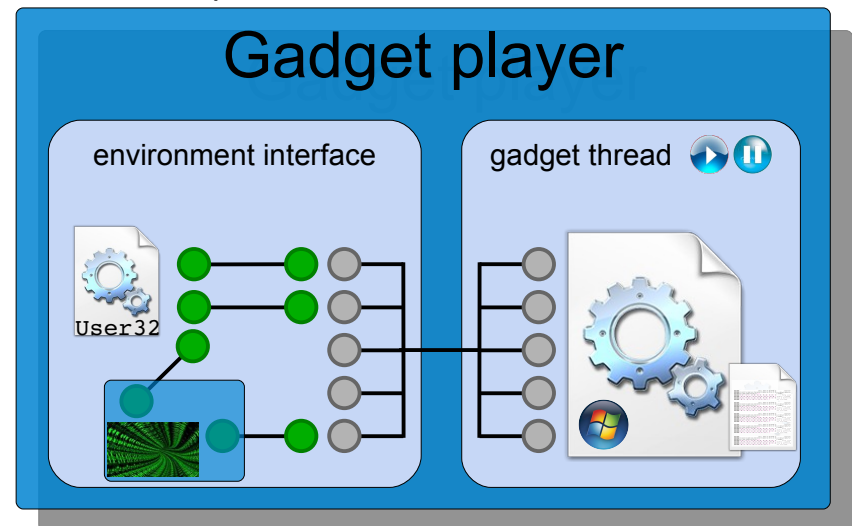
- Player can use gadget as *transformation oracle*
 - *input* is transformed into *output*, depending on algorithm implemented by gadget
 - example: sample reads local data, obfuscates, and transmits to remote host
- In many scenarios, the inverse algorithm would be interesting, however
 - we capture obfuscated traffic and want the plain-text data that has been transmitted
- In the paper, we present basic inversion capabilities:
 - *detect* (taint) *dependencies* between bytes in input and output
 - apply guided *brute-force* heuristics to invert algorithm contained in the gadget

So ... again, why...?

Gadget Benefits

Int. Secure Systems Lab
Vienna University of Technology

- ... so why not simply execute it in a VM (over and over again)?
 - *sleep timeouts*: can be eliminated during gadget extraction
 - *fast & lightweight analysis*:
 - no virtual environment, snapshot restoring
 - *we ran our analysis on Linux (under Wine)!*
 - *precise, uncluttered behavior observation*
 - *advanced monitoring*: the player has access to the gadget's heap and stack regions!
 - *environment tampering*: all requests go through a single interface: tamper with date or time, registry settings, hostOS, remote hosts contacted, ...



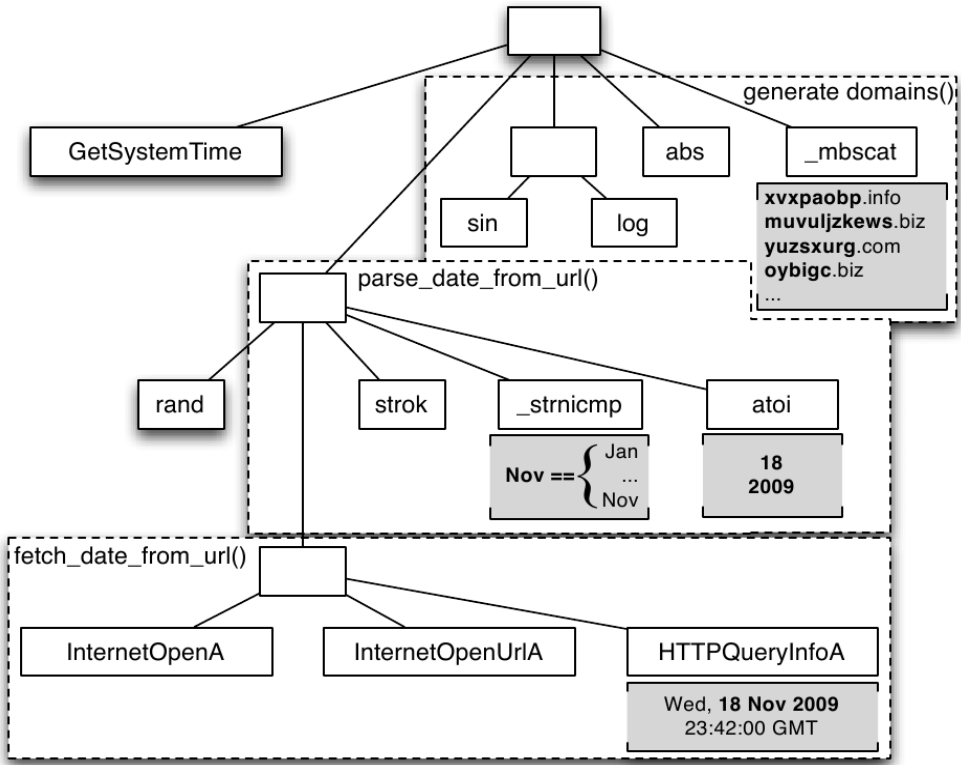
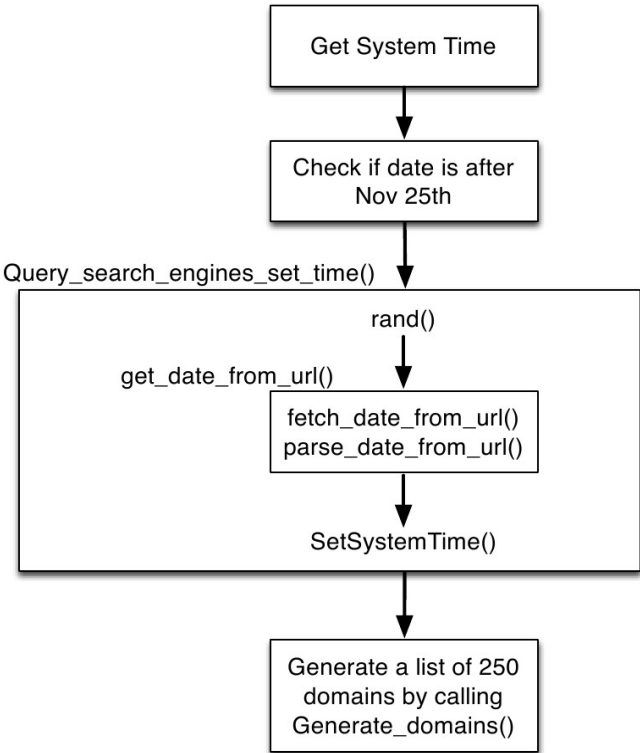
Gadget examples

Conficker DGA

Int. Secure Systems Lab
Vienna University of Technology

- *Conficker* generates (pseudo) random domain names upon startup
- Current time, fetched from remote site (e.g., msn.com), controls domain generation randomization seed
- Randomly selected domain name is used for contacting C&C host
- Gadget:
 - start extraction (slicing) from invocation of `DnsQuery_W`
 - extracts complete *Domain Generation Algorithm* (DGA)
 - see *one domain* on query invocation
 - find *all domains* on gadget heap
- Possible environment tampering:
 - manipulate remote site's reply to change DGA input (i.e., date for which domains are generated)

Conficker DGA



P. Porras et al. "A Foray into Conficker's Logic and Rendezvous Points"

Inspector Gadget control flow "debug output"

Cutwail Spam Templating

Int. Secure Systems Lab
Vienna University of Technology

- *Cutwail* (mass-mailer) generates spam Emails from *templates* downloaded from remote C&C hosts
- Communication employs proprietary encryption algorithm
- Template is not stored on file system
 - content decrypted and handled solely in memory
- Gadget:
 - inspect download behavior
 - start extraction after download is complete
 - *Inspector* suggests to automatically refine extraction starting-point to *end-of-decryption*
 - extract complete template download & decryption algorithm

Cutwail Spam Templating

Int. Secure Systems Lab
Vienna University of Technology

```
"{_FIRSTNAME} {_LASTNAME}" <{MAIL_FROM}>
```

```
Hello my new friend, I search a good man at other  
country...\n For me it to communicate for the first  
time with the person from other country, by  
Internet.\nAnd it  
...
```

```
{\nReceived}
```

```
Message-ID: <{DIGIT[10]}.{SYMBOL[8]} {DIGIT[6]} @{nHOST}>  
From: {TAGMAILFROM}  
To: <{MAIL_TO}>  
Subject: {SUBJECT}  
Date: {DATE}  
MIME-Version: 1.0  
Content-Type: multipart/mixed;  
boundary="-----_NextPart_000_0006_{_nOutlook_Boundary}"  
X-Priority: 3  
X-MSMail-Priority: Normal  
X-Mailer: Microsoft Outlook Express {_nOutlookExpress_4}
```

Cutwail Spam Templating

Int. Secure Systems Lab
Vienna University of Technology

```
configver      194
addr           91.206.231.230
port           25
knockdelay     60
mxrecvtimeout 120
mxconntimeout 120
maxtrybadfrom  1
maxtryconn     5
```

```
nHOST
{nChar[5-15]}.{nChar[5-15]}
  .{LET:ru,org,com,va,net,biz,
    info,tv,ua,su}

nReceived
Received: from [{nIP}]
  (helo={nHOST})
  {N[1]}by {BOT_HOST}
  ...
```

```
FIRSTNAME      Christi
                Lea
                Staci
                Jodie
                Summer
                Katharine
                ...

LASTNAME        Schafer
                Stacy
                Grayson
                Ham
                Landers
                Mims
                Parham
                Pritchett
                ...

name            Lusia R., Texas
                Lusia R., New York
                Lusia R., Chicago
                Lusia R., Colorado
                Lusia R., Boston
                Lusia R., Washington
                Lusia R., Las Vegas
                Lusia R., Bellevue WA
                Lusia R., San Diego
                Amelia B., Chicago
                ...
```

URLZone Config Update

Int. Secure Systems Lab
Vienna University of Technology

- *URLZone* (BHO banking-trojan) sniffs and manipulates user interaction with banking web-site
- Steals credentials and hides previous (malicious) transactions from user
- Remote configuration through encrypted configuration files
 - domains to attack
 - URLs to inspect
 - form content to modify
- Gadget:
 - extract complete template download & decrypt algorithm
 - similar to *Cutwail* gadget

URLZone Config Update

Int. Secure Systems Lab
Vienna University of Technology

```
=====POST=====
[ ITBEGINBLOCKHOOK ]
ITHOST= |banking.postbank.de |End
ITPAGE= | /app/login.d* |End
ITMETHOD= | 2 |End
ITIFINIT= | %DISP% |End
ITREQMATH= | jsOn=* &accountNumber=* &pinNumber=* |End
```

```
----- STATA -----
ITINJHOST= |my.hypovereinsbank.de |End
ITINJPAGE= | /*?view=/* |End
...
ITINJSTART= |Aktueller Kontosaldo</label>[*]
              <p class="right">|End
ITINJEND= |</p>|End
ITINJCODE= | |End
ITINJPASTE= | %HYPOBAL%+%AMOUNT%-%TRUEAMOUNT% |End
ITINJPASTEMN= |<span
class="negative-balance">%HYPOBAL%+%AMOUNT%-
              %TRUEAMOUNT%</span><span
class="negative-balance">EUR</span>|End
```

Summary

Int. Secure Systems Lab
Vienna University of Technology

- Dynamic analysis is resource consuming, results are cluttered and limited to temporary snapshots of malicious behavior
- *Inspector* allows to automatically extract behavior into standalone *gadgets*
- Gadgets can be reused in many scenarios and
 - enhance information extraction
 - simplify repeated analysis of behavior
- Evaluation shows that extraction is applicable to real world, malicious programs

Thanks for listening!

