



# Hacking the Hacker

## It's Payback Time...

Joel Eriksson

Eriksson

04/11/08 | Session Code: RR-401

**RSA**CONFERENCE**2008**

## Background

- Targeted attacks are getting more common
- Often exploits client side vulnerabilities
- Usually installs a trojan of some kind



## Background

- Once inside, may spread further as a worm
- Usually connects out to the attacker
- May connect through proxies/TOR

Purely hypothetical...

- Your network is under attack
- Thousands of infected clients/servers
- A Remote Administration Trojan (a RAT) is used



## What do you do?

- Pull the plug, spend months on cleaning up
- Pretend it never happened
- Get even...

## Getting even

- Determine the following:
  - What information has been stolen
  - The identity of the attacker(s)
  - His/her/their objectives
- Take appropriate countermeasures

# Is this even possible?

- It depends!
- Recording all network traffic helps...
- Get in touch with a skilled reverse-engineer

## What's reversing got to do with it?

- Need to determine the capabilities of the RAT
- See where the RAT is connecting
- Reverse the protocol

## Why reverse the RAT protocol?

- To create NIDS-signatures for detecting it
- Simulate the server, send uninstall command!
- Analyze packet-dumps, see what has been done

## What about finding the hacker?

- That's a tough one...
- Probably using proxies/TOR
- Any ideas?

## Case study 1 – Bifrost (AKA Bifrose)

- Public RAT
- Made in Sweden
- Used internationally – English GUI
- Has been used in targeted attacks
- For examples, google: bifrose targeted attacks

## Bifrost v1.1.02

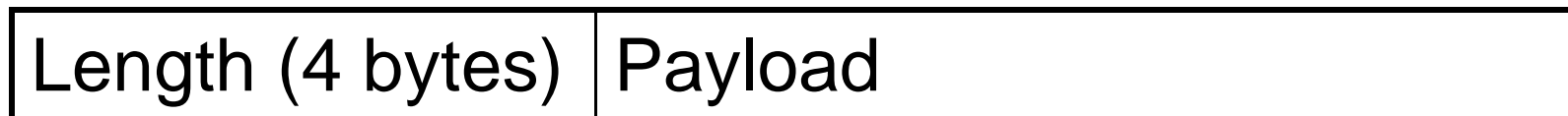
- Released in 2005
- Specialized in bypassing firewalls
- Was originally researched by me in 2006

## Bifrost v1.1.02

- Packed to make static analysis harder
- Manually unpacked using OllyDbg and ImpREC
- Unpacked binary analyzed further with IDA Pro

## Bifrost v1.1.02

- Uses a simple protocol



- Payload is XOR-encrypted
- Static key:

A3 78 26 35 57 32 2D 60 B4 3C 2A 5E 33 34 72 00

## Bifrost v1.1.02

- Payload consists of



- Parameters usually separated by '|'
- Binary data as parameters for some commands

## Bifrost v1.1.02

- The first command sent registers the client
- Includes local IP, username, hostname, etc
- Many commands contain responses to requests from the attacker, such as getting a listing of processes, files or a screencapture

## Bifrost v1.1.02

- Initial attack surface: The register command
- Or so I thought...

## Bifrost v1.1.02

- Accepts responses to requests never sent!
- Even when the register command has not been sent, so the connected client is not visible in the GUI, it accepts responses to any command. 😊
- Suddenly a much larger attack surface

## Bifrost v1.1.02

- Built a very simple fuzzer
- Sent random commands, with random payloads
- Crashed after a certain sequence of commands
- Corrupted heap...

## Bifrost v1.1.02

- Initial crash probably due to dangling pointer
- Further analysis with IDA revealed another issue, in handling the commands in question

```
0041FBD3    mov  eax, [esi+120h]
0041FBD9    add  eax, 5000
0041FBDE    push eax
0041FBDF    call _malloc
```

## Bifrost v1.1.02

- Integer-overflow, leading to heapbased overflow
- Exploit developed as an MSF2 module

## Bifrost v1.1.02

- Exploit overwrites the UnhandledExceptionFilter IAT-entry, which will be called when Bifrost would have otherwise crashed due to a corrupted heap 😊
- The IAT entry would normally have been read-only, but the packer does not set it read-only after restoring the IAT

Bifrost v1.1.02

**Demo**

bitsec

## Bifrost v1.2.1

- Released in 2007
- Latest, and last, public release
- Was researched while preparing for this talk
- Built-in support for jumping through proxies, and even TOR with a plugin

## Bifrost v1.2.1

- Old exploit didn't work anymore
- Turned out the protocol had changed, a bit

## Bifrost v1.2.1

- New protocol



- Payload is RC4-encrypted
- Static key:

A3 78 26 35 57 32 2D 60 B4 3C 2A 5E 33 34 72 00

- XOR had simply been replaced with RC4...

## Bifrost v1.2.1

- Updated exploit to use RC4, still no effect
- Managed to find out why after spending some time in IDA Pro, had to send a new command to trigger the actual buffer overflow
- The same old integer-overflow was still there
- Exploit developed as a shellscript using tools made in C for RC4-encoding the commands and their payloads

Bifrost v1.2.1

**Demo**

bitsec

## Case study 2 – PCShare (AKA Pcclient)

- Semi-public RAT
- Made in China
- Only chinese GUI available
- Has been used in targeted attacks
- Was originally researched by me in 2006

## PCShare

- Packed to make static analysis harder, just as Bifrost and most other RATs / malware
- Manually unpacked using OllyDbg and ImpREC
- Unpacked binary analyzed further with IDA Pro

## PCShare

- Simple HTTP-based protocol in the top layer

**GET /index.asp?<Command><Parameters>**

- Command = 4-digit number
- Parameters = Hex-encoded

## PCShare

- Command sent to register the client = 5021
- Connection kept open, the server sends requests (in binary) through this connection
- Responses sent as new HTTP-requests

## PCShare

- Unlike Bifrost, does NOT accept responses to requests that have not been sent
- Initial attack surface – The 5021-command
- Did not find anything interesting...

## PCShare

- Have to go one step further and simulate a client, to be able to attack other commands as the hacker is attempting to control his victim

## PCShare

- The default action, when the hacker double-clicks on the victim of interest, is to bring up a filemanager
- This is handled in a new process and by a new binary, pretty much like a CGI

## PCShare

- In the file manager the hacker may, for instance:
  - Upload files to the victim
  - Download files from the victim
  - Download and execute files from the victim
- Download and execute seems interesting ;)

## PCShare

- Does not accept a file from the victim unless a download is actually in progress
- Or at least, that's how it is supposed to work ;)
- Used IDA Pro to see exactly what happens

## PCShare

- To be able to update a progressbar during the download, the window handle of the progressbar is sent to the victim when a download starts
- The filename, file contents and the window handle of the progressbar is sent from the victim to the hacker

## PCShare

- The only requirement for accepting the response to the download-and-execute command is that a valid window handle is included with it 😊
- Does not check that the window handle actually belongs to a progressbar, or that the hacker has even sent a download request
- Bruteforcing a valid window handle is easy
- Hmm, who's the victim now? ;)

## PCShare

- Two exploits
- One simulates the victim, handling some of the commands, but still pretty much depends on the hacker only double-clicking and bringing up the file manager and not trying to do anything more advanced
- One passively sniffs a real victim (or rather, a "honeypot" victim) and sends the attack whenever the hacker brings up the file manager

PCShare

**Demo**

bitsec

## Summing it up

- Malware-analysis got a lot more entertaining 😊
- A specially prepared NIDS may be used for:
  - Detecting a victim connecting to the hacker
  - Automatically redirecting the victim to a simulated server
  - Sending the uninstall command to the victim
  - Launching an attack on the hacker, with a custom payload to determine his real IP address and other useful information, such as the MAC address etc
- Might also be an interesting way to deal with botnets and DDOS attacks ;)

## Legal disclaimer

- Only been doing this as pure research so far
- Actually launching the attack, even with a custom payload used only for extracting certain information, is probably still a very dark grey area...
- But when national security is at stake, perhaps dark grey is quite ok. ;) Who am I to judge...