



Analyze cyber threats faster. Together.

Lars Haukli
lars@flip.re



~~I am~~ I used to work as a
reverse engineer



Storm Worm (2007)



Surface (quick static)



Surface (quick static)
Dynamic (run in VM)



Surface (quick static)
Dynamic (run in VM)
Static (code analysis)



Dynamic analysis is key to
effective threat analysis



Relied on debuggers



Relied on debuggers
Step through packers



Relied on debuggers
Step through packers
Locate payload



Relied on debuggers
Step through packers
Locate payload
Dump memory



Relied on debuggers
Step through packers
Locate payload
Dump memory
Fix imports



Relied on debuggers
Step through packers
Locate payload
Dump memory
Fix imports
Load in IDA



During incident handling
you are on a clock



Storm Worm did not run in
my lab so I was stuck



Toni Koivunen *via* IRC



```
mov eax, 0x564d5868
```



```
mov eax, 0x564d5868  
mov ebx, 0
```



```
mov eax, 0x564d5868
```

```
mov ebx, 0
```

```
mov ecx, 0x0a
```



```
mov eax, 0x564d5868  
    mov ebx, 0  
    mov ecx, 0x0a  
mov dx, 0x5658
```



```
mov eax, 0x564d5868  
    mov ebx, 0  
    mov ecx, 0x0a  
    mov dx, 0x5658  
    in eax, dx
```



```
mov eax, 0x564d5868
    mov ebx, 0
    mov ecx, 0x0a
    mov dx, 0x5658
    in eax, dx
cmp ebx, 0x564d5868
```



Collaboration helped me
immediately understand



Threat analysis is hard



Can we make it easier?



Iterative threat analysis



Start with a quick static analysis



Refine *your* static analysis
with events and traces from
dynamic analysis



Tweak *your*
dynamic analysis based on
static code analysis



As you make new
discoveries, iterate to
improve your analysis as
needed



Extract actionable
intelligence, and share
initial conclusion



Collaborate with others to
understand more, faster.



The *technology* stack.



Next.js

React framework with
Rust-based javascript
tooling



Radix UI

Unstyled UI component
library + icons + colors



Monaco Editor

The code editor that
powers Visual Studio Code



XTerm.js

Terminals in the browser!



Rocket.rs

Web framework for Rust



QEMU + KVM Virtualization



Frida

Frida server running in VM
+ Core SDK integrated into
Rocket.rs backend



radare2

Reverse engineering framework



Ghidra

Decompiler



Hyperflip

Hypervisor-level debugging



Enjoy the demo.



Thank you.

Join us on Discord via
<https://flip.re>

Lars Haukli
lars@flip.re