DEBUGGING & MONITORING C2 TRAFFIC WITH HAKA

BENOIT ANCEL & MEHDI TALBI
STORMSHIELD

vb 2016
DENVER
5 - 7 October 2016

OUTLINE

- Motivations
- Haka
- Malware monitoring - Demo
  - China-Z
  - Athena
- Conclusions

MOTIVATIONS

- Long-term malware monitoring
  - Threat Intelligence
  - Tracking malware campaign

- Lack of means to monitor, debug and monitor malware network activities

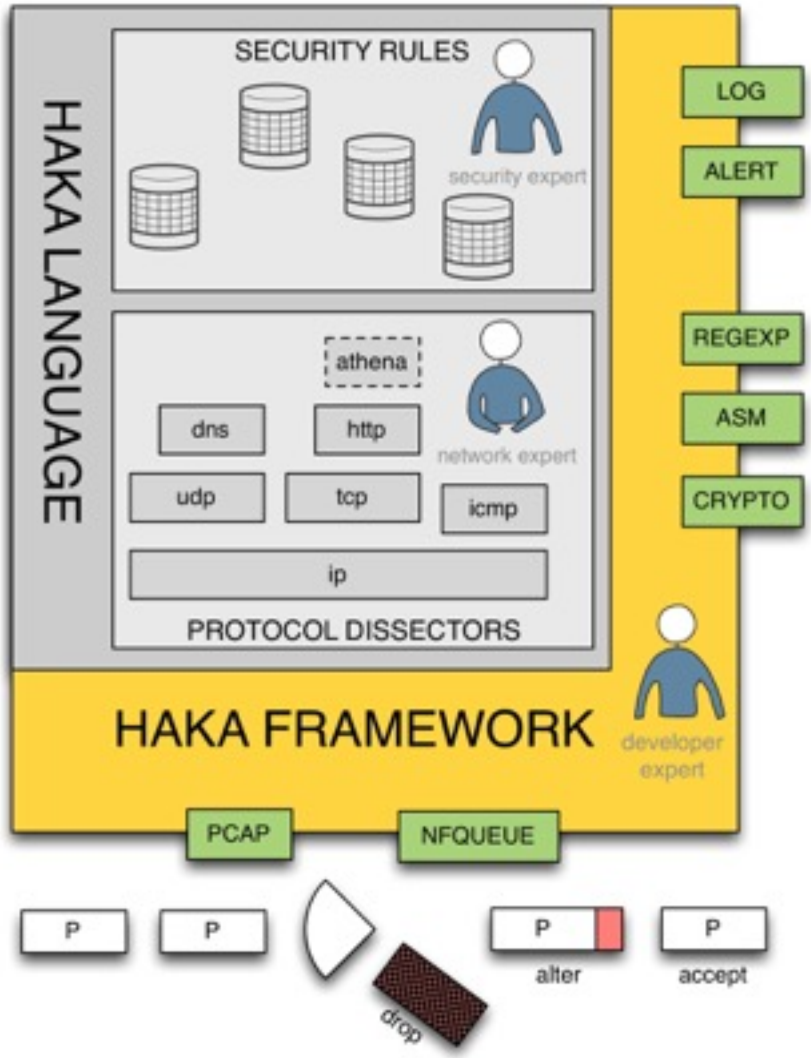
HAKA

AN OPEN SOURCE SECURITY-ORIENTED LANGUAGE

HAKA
MOTIVATIONS

1. Malware protocol dissection
2. Advanced API for packet and stream manipulations
3. On-the-fly packet modification (hijack botnet commands)
4. Interactive packet filtering mode (break into packets and inspect their content)
5. Instruction disassembler (disas. packet content at network level)
6. Dedicated tool to monitor traffic : Hakabana

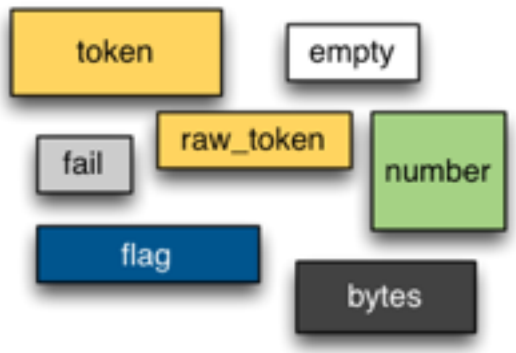
HAKA ARCHITECTURE

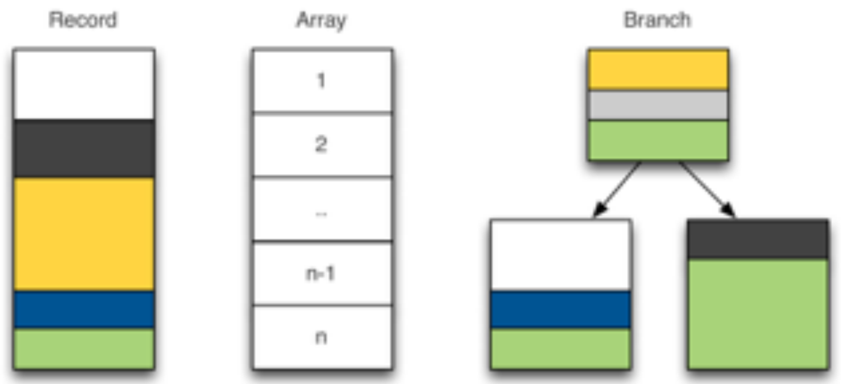


HAKA
PROTOCOL DISSECTION

- Protocol Specification : Message format + State machine
- Support of text-based protocols (http) and binary-based protocols (dns)
- Support of packet-based protocols (icmp) and stream-based protocols (http)
- All parsed fields are available in read/write access to security rules

HAKA
GRAMMAR BUILDING BLOCKS

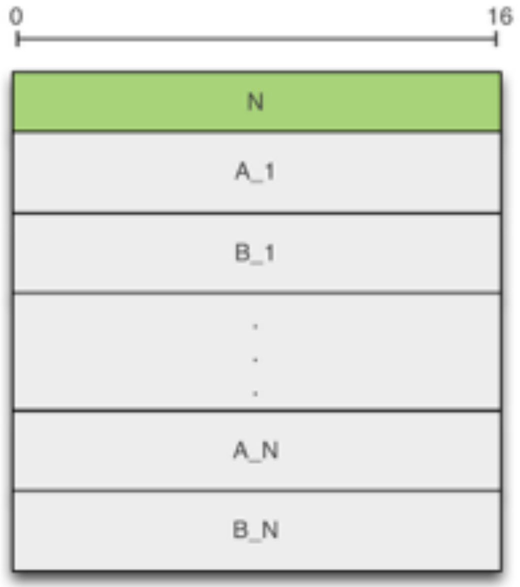


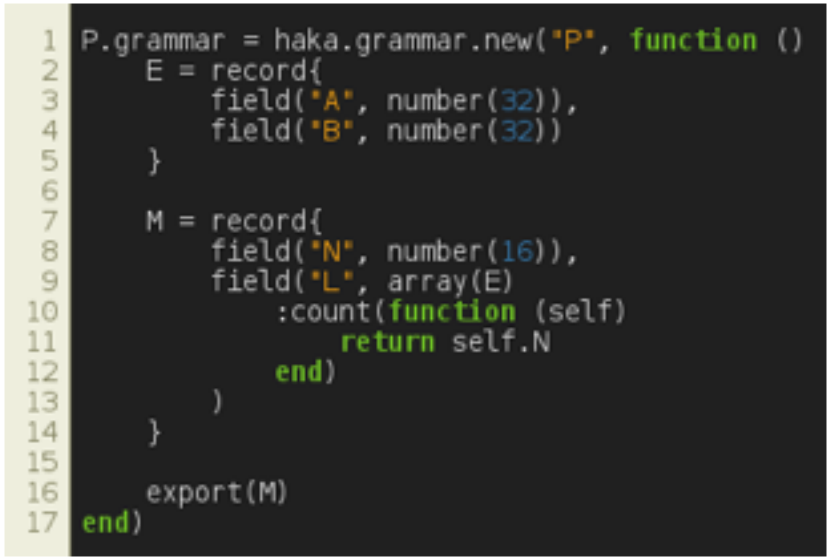


basic blocks

compound blocks

HAKA
PROTOCOL DISSECTION EXAMPLE



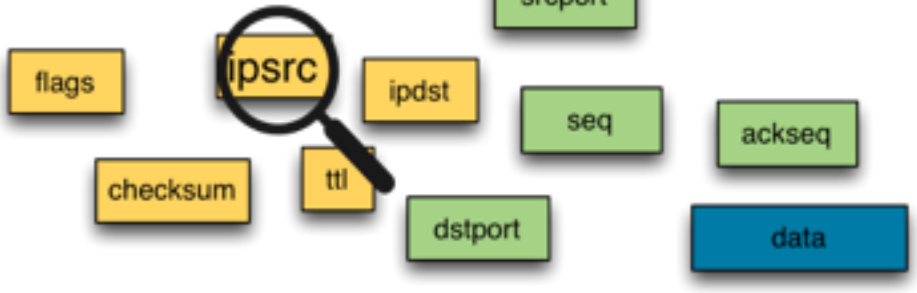


```
P.grammar = haka.grammar.new("P", function ()
    E = record{
        field("A", number(32)),
        field("B", number(32))
    }

    M = record{
        field("N", number(16)),
        field("L", array(E)
            :count(function (self)
                return self.N
            end)
        )
    }

    export(M)
end)
```

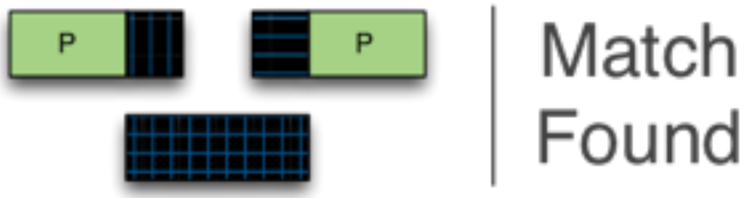
STORMSHIELD

HAKA
SECURITY RULES (1)

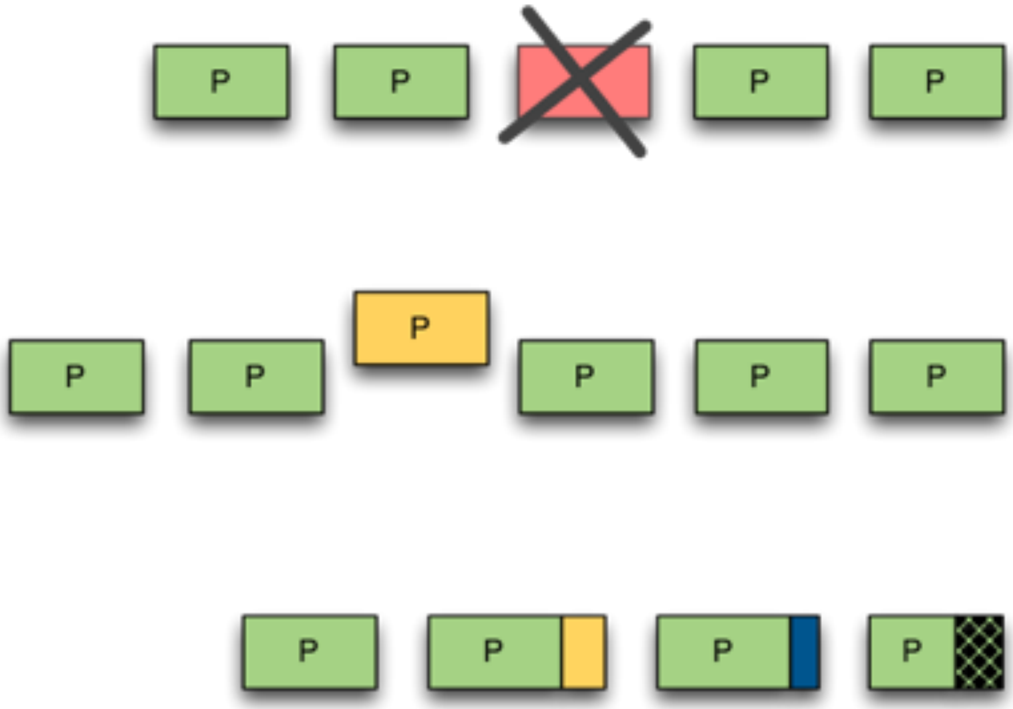
- Inspecting packet content



- Matching a malicious pattern across multiple packets



HAKA
SECURITY RULES (2)

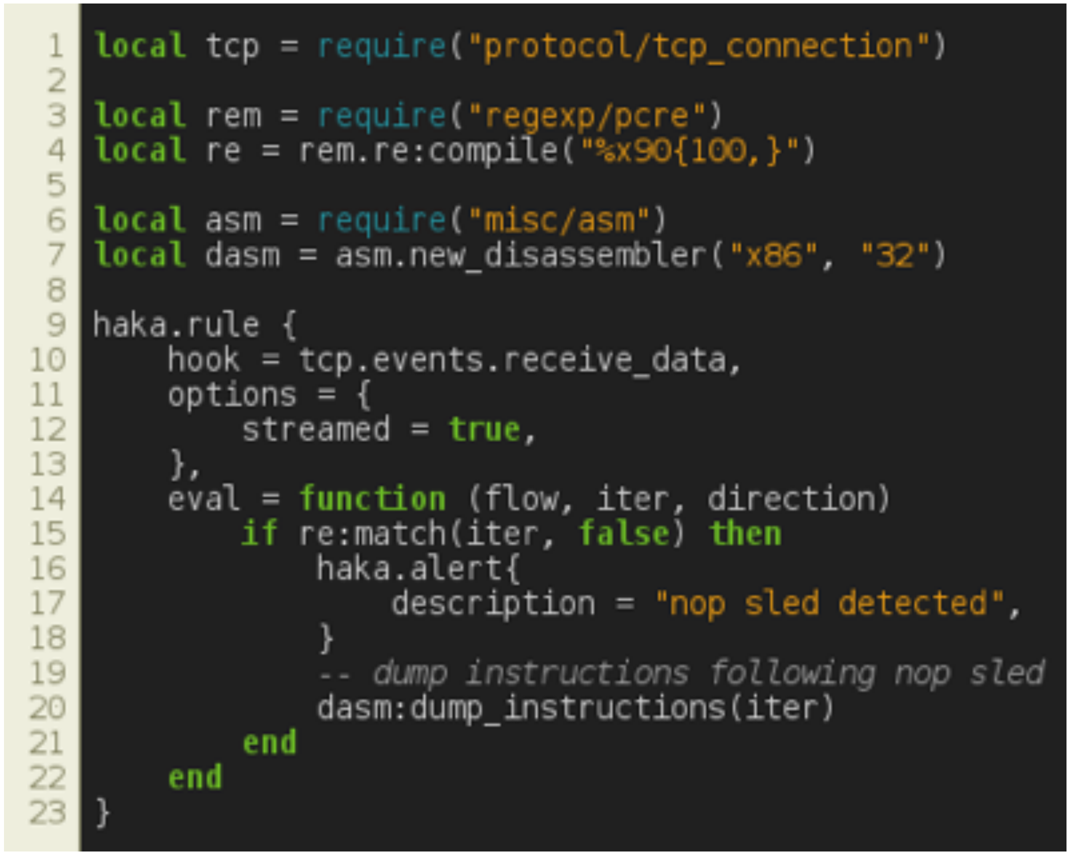


drop packets

inject packets

alter packets

HAKA
SECURITY RULE EXAMPLE



```lua
local tcp = require("protocol/tcp_connection")

local rem = require("regexp/pcre")
local re = rem.re:compile("%x90{100,}")

local asm = require("misc/asm")
local dasm = asm.new_disassembler("x86", "32")

haka.rule {
    hook = tcp.events.receive_data,
    options = {
        streamed = true,
    },
    eval = function (flow, iter, direction)
        if re:match(iter, false) then
            haka.alert{
                description = "nop sled detected",
            }
            -- dump instructions following nop sled
            dasm:dump_instructions(iter)
        end
    end
}
```

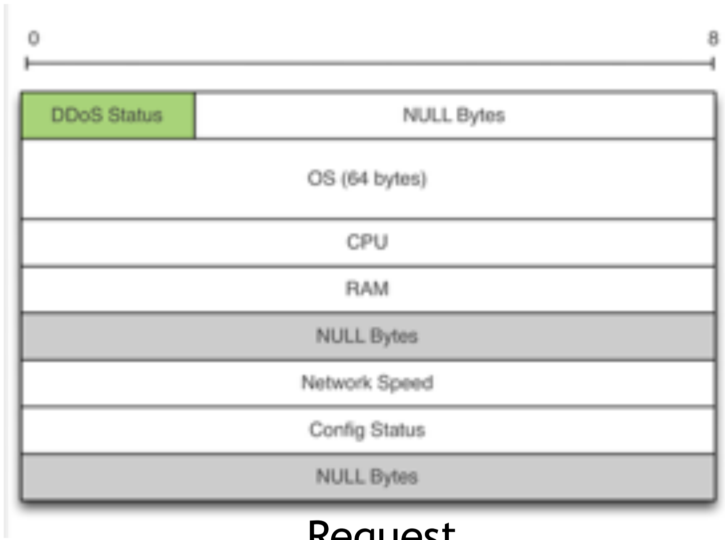
MONITORING C2 TRAFFIC

USE CASES

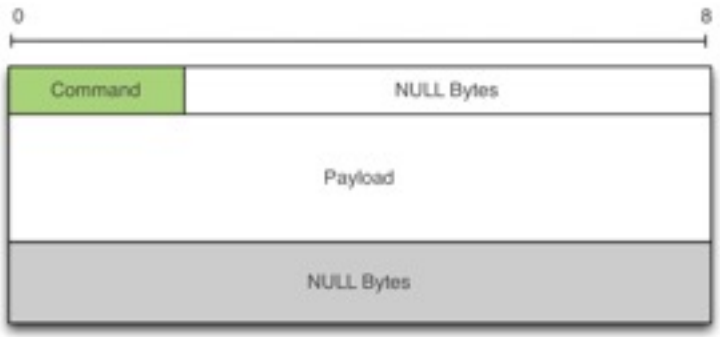
CHINA-Z OVERVIEW

- Well known Chinese botnet
- 5 known variants: China-Z/{A, B, C, O, S}
- Includes binary and configuration update features
- DDoS campaigns targeting Asia

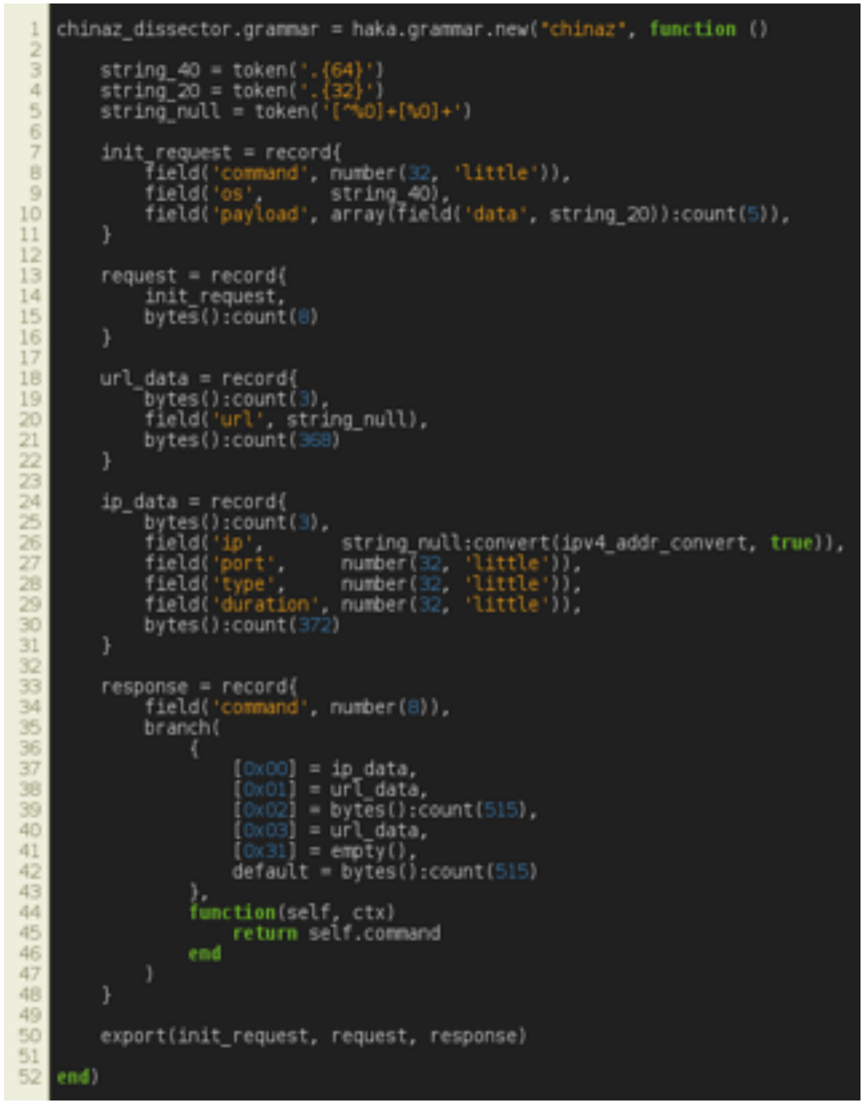
CHINA-Z
C2 COMMUNICATIONS



Request



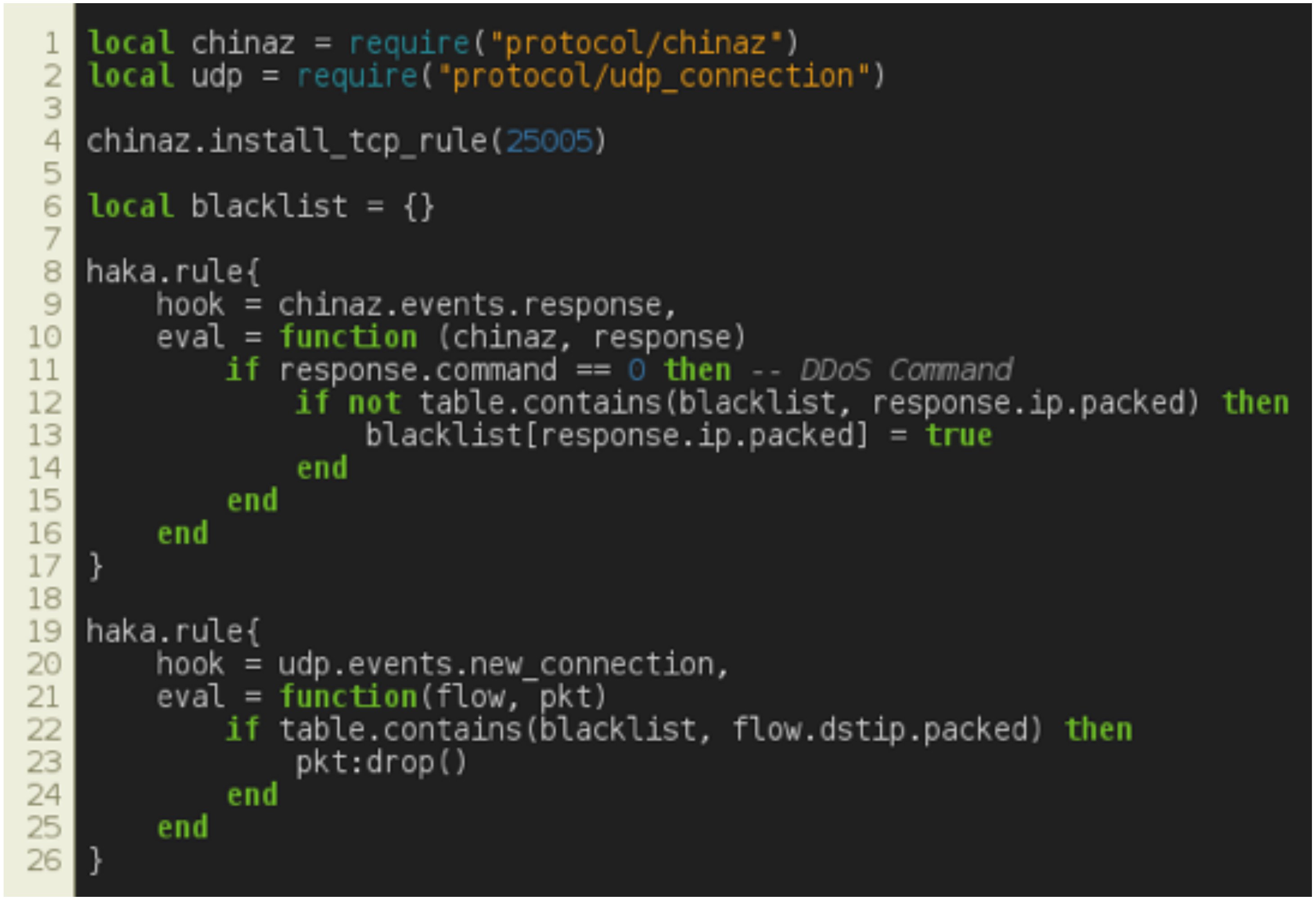
Response



```
chinaz_dissector.grammar = haka.grammar.new("chinaz", function ()

    string_40 = token('.{64}')
    string_20 = token('.{32}')
    string_null = token('[^%0]+[%0]+')

    init_request = record{
        field('command', number(32, 'little')),
        field('os',      string_40),
        field('payload', array(field('data', string_20)):count(5)),
    }

    request = record{
        init_request,
        bytes():count(8)
    }

    url_data = record{
        bytes():count(3),
        field('url', string_null),
        bytes():count(368)
    }

    ip_data = record{
        bytes():count(3),
        field('ip',       string_null:convert(ipv4_addr_convert, true)),
        field('port',     number(32, 'little')),
        field('type',     number(32, 'little')),
        field('duration', number(32, 'little')),
        bytes():count(372)
    }

    response = record{
        field('command', number(8)),
        branch(
            {
                [0x00] = ip_data,
                [0x01] = url_data,
                [0x02] = bytes():count(515),
                [0x03] = url_data,
                [0x31] = empty(),
                default = bytes():count(515)
            },
            function(self, ctx)
                return self.command
            end
        )
    }

    export(init_request, request, response)

end)
```

STORMSHIELD

CHINA-Z - SECURITY RULES BLOCKING DDOS ATTACKS



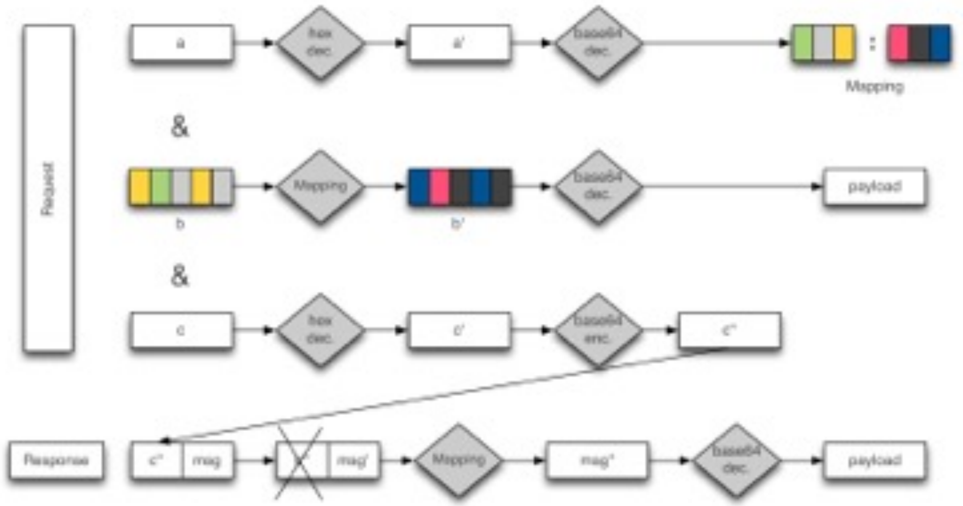
```
local chinaz = require("protocol/chinaz")
local udp = require("protocol/udp_connection")

chinaz.install_tcp_rule(25005)

local blacklist = {}

haka.rule{
    hook = chinaz.events.response,
    eval = function (chinaz, response)
        if response.command == 0 then -- DDoS Command
            if not table.contains(blacklist, response.ip.packed) then
                blacklist[response.ip.packed] = true
            end
        end
    end
}

haka.rule{
    hook = udp.events.new_connection,
    eval = function(flow, pkt)
        if table.contains(blacklist, flow.dstip.packed) then
            pkt:drop()
        end
    end
}
```

CHINA-Z
DEMO

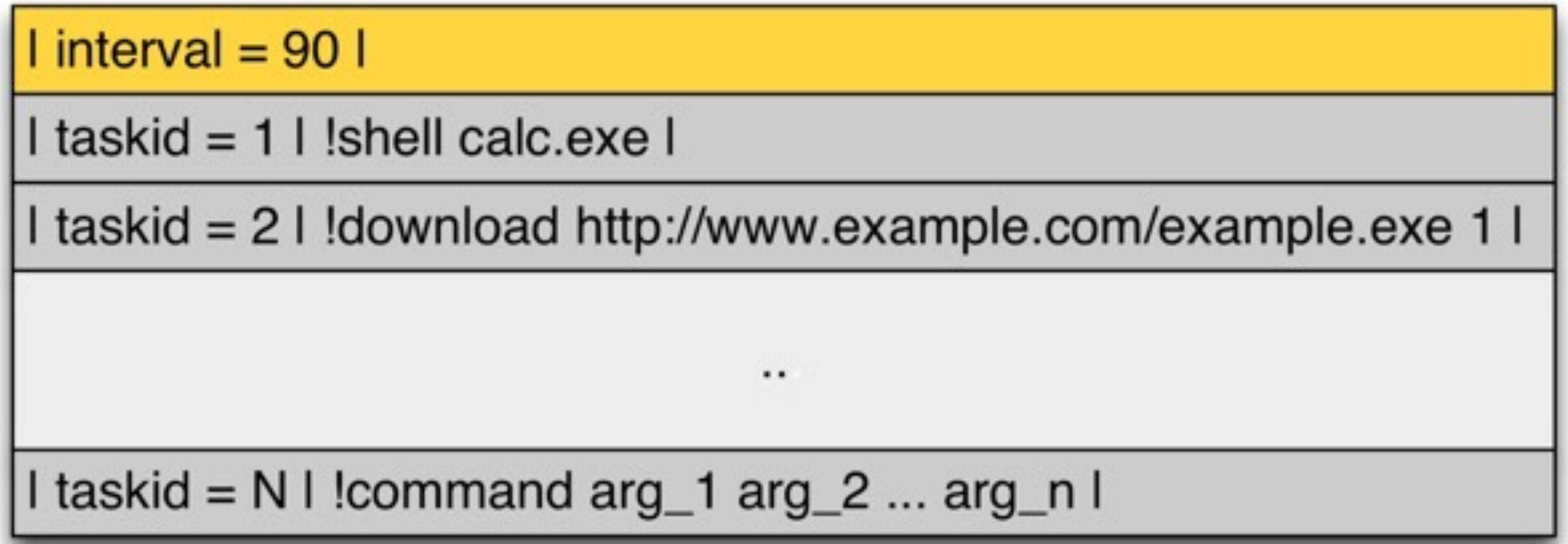
https://www.youtube.com/watch?v=-XeyMMcZ-TI

ATHENA OVERVIEW

- Several features : click fraud, ddos, bot killer, etc.
- Communicates over http
- Encoded requests and responses (hex + base64 encodings)

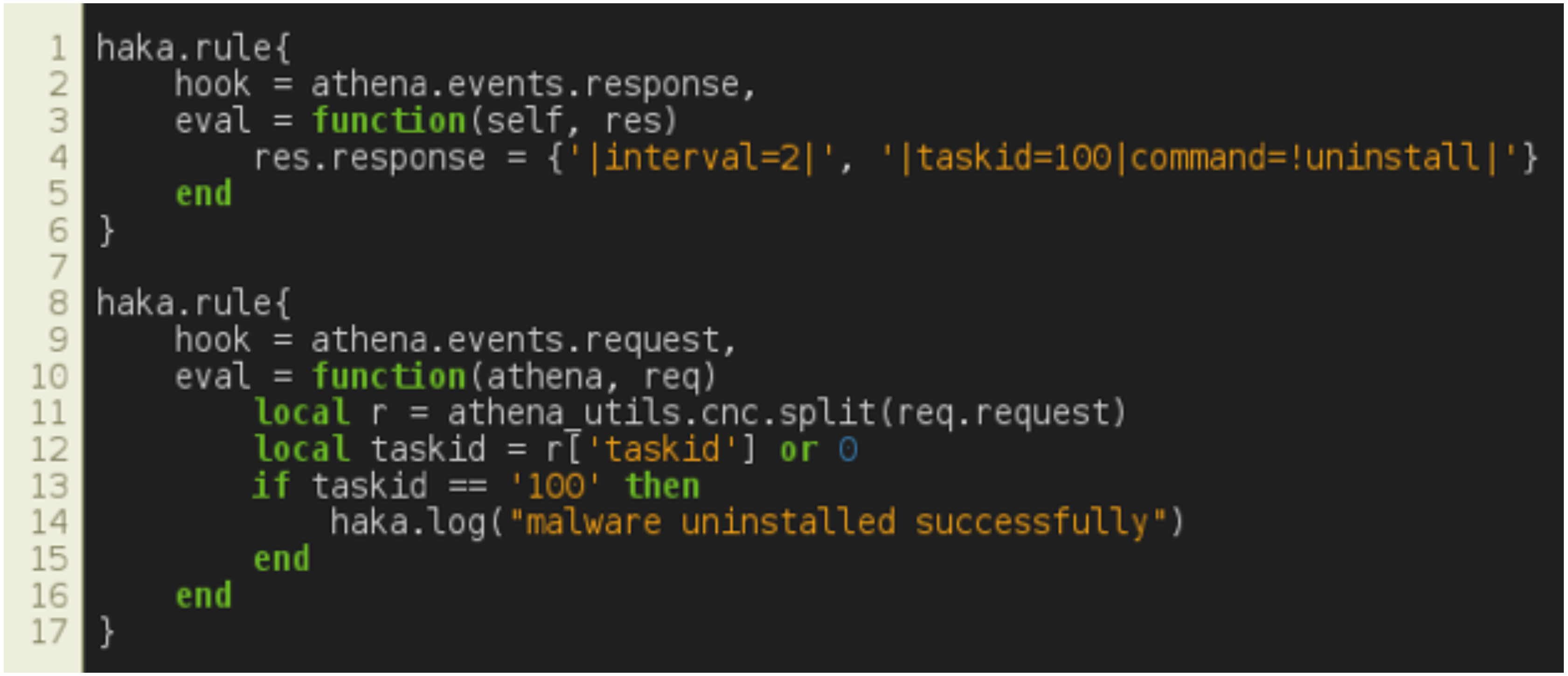


ATHENA COMMANDS



```
| interval = 90 |
| taskid = 1 | !shell calc.exe |
| taskid = 2 | !download http://www.example.com/example.exe 1 |
..
| taskid = N | !command arg_1 arg_2 ... arg_n |
```

ATHENA - SECURITY RULES
HIJACKING BOTNET COMMANDS



```
haka.rule{
    hook = athena.events.response,
    eval = function(self, res)
        res.response = {'|interval=2|', '|taskid=100|command=!uninstall|'}
    end
}

haka.rule{
    hook = athena.events.request,
    eval = function(athena, req)
        local r = athena_utils.cnc.split(req.request)
        local taskid = r['taskid'] or 0
        if taskid == '100' then
            haka.log("malware uninstalled successfully")
        end
    end
}
```

STORMSHIELD

ATHENA
DEMO

https://www.youtube.com/watch?v=5hMKN0k_zCQ

CONCLUSIONS

- Haka provides several features to monitor and debug C2 traffic.

- Future works : provide a repository of malware protocol dissectors

THANK YOU

github.com/haka-security

@hakasecurity

haka-security.org