



GUARD SCRIPT

A PROGRAMMING LANGUAGE
CRAFTED TO AID
CYBERSECURITY
PROFESSIONALS IN THEIR
TASKS



MOTIVATION

**GUARDSRIPT IS
A PROGRAMMING LANGUAGE
ENGINEERED TO EMPOWER
CYBERSECURITY
PROFESSIONALS WITH ITS
STRAIGHTFORWARD AND
CONCISE SYNTAX.**


**IT WAS DESIGNED WITH THE
AIM OF SIMPLIFYING
CYBERSECURITY OPERATIONS
INTO A SINGLE, CENTRALIZED
AND SIMPLIFIED PARADIGM.**



CHARACTERISTICS

SIMPLICITY AS ITS CORE

WITH A STRAIGHTFORWARD SYNTAX,
WHICH RESEMBLES PYTHON AND
GOLANG, GUARDSCRIPT AIMS TO
FACILITATE THE TRANSITION OF
PROFESSIONALS ADAPTED TO THE
LANGUAGES MOST USED IN THE
SECURITY ENVIRONMENT.



```
1  int:x = 5
2  int:y = 5
3  int:z
4  z = (x+y - x/y + x)/3
5  if z>3 && x == 5{
6      show("Teste1")
7  }else{
8      show("Teste2")
9  }
10 show(z)
```

EASILY INTEGRATED WITH EXECUTABLE ANALYSIS

WITH BUILT IN CAPABILITIES,
GUARDSCRIPT TURNS EASIER THE
ANALYSIS AND DETECTION OF
MALWARES IN THE MACHINE,
IN APPROACH SIMILAR TO YARA
RULES.

WITH RULE TYPE AND
MATCH() FUNCTION, YOU CAN DETECT
STATICAL STRINGS IN EXECUTABLES.

```
1 str:file_name = "../LexParser/parser"
2 rule:rules_ryuk {
3   str:a = ".php?",
4   str:b = "uid=",
5   str:c = "&uname=",
6 }
7 if match(file_name, rules_ryuk){
8   show("Arquivo suspeito")
9 } else{
10  show("Arquivo normal")
11 }
```

EFFECTIVE FOR NETWORK TRAFFIC ANALYSIS

GUARDSCRIPT FACILITATES SIMPLE
TRAFFIC ANALYSIS WITHOUT THE
NEED FOR EXTERNAL TOOLS OR
DEEP KNOWLEDGE OF SOCKET
COMMUNICATIONS.

TRAFFIC_INFORMATION() RETURNS
TCP PACKETS EXCHANGED DURING
ITS RUNTIME.

```
1  int:initial
2  int:final
3  show("Starting Monitoration")
4  show("=====")
5  foreach initial = 1 to final = 10{
6      traffic_information()
7  }
8  show("=====")
9  show("Report complete")
```

```
0  show("Report complete")
```

```
0  show("=====")
```


SMOOTH FOR SCANNING HOSTS

IDEAL FOR SCANNING HOSTS OR IP ADDRESSES ON DIFFERENT PORTS AND USING THIS INFORMATION DIRECTLY IN YOUR SCRIPT, WITHOUT THE NEED FOR EXTERNAL INTEGRATION.

```
1 show("Digite o endereço de ip para verificar: ")
2 str:ip_address = input()
3 int:port_initial
4 int:port_final
5 int:port = 8079
6 foreach port_initial = 8079 to port_final = 8082{
7     if scanhost(ip_address,port) == "open"{
8         show("Porta aberta: ")
9         show(port)
10    } else{
11        show("Porta fechada: ")
12        show(port)
13    }
14    port = port + 1
15 }
```

The image features a central digital padlock with intricate circuit board patterns in yellow and green. The padlock is set against a dark blue background filled with floating binary digits (0s and 1s) and glowing network lines. On the far left, there is a vertical orange bar with a white dotted pattern. The word "CURIOSITIES" is written in a large, white, serif font across the middle of the image.

CURIOSITIES



INSPIRATION

THE CENTRAL IDEA FOR GUARDSCRIPT ORIGINATED FROM MY UNDERGRADUATE RESEARCH, DURING WHICH, WHILE ENGAGING WITH MALWARE ANALYSES AND ATTACK TOPOLOGIES, I IDENTIFIED THE NEED FOR MORE ACCESSIBLE AND ADAPTABLE TOOLS.

BUILT-IN FUNCTIONS

THE BUILT-IN FUNCTIONS ESSENTIALLY CONSIST OF INTEGRATIONS USING LOW-LEVEL SOCKET CONNECTIONS FOR TRAFFIC MONITORING (WHICH REQUIRE ADMINISTRATOR PRIVILEGES) AND PORT SCANNING, IN ADDITION TO AN EMBEDDED YARA MODULE FOR FACILITATING MALWARE DETECTION AND SUBSEQUENT COUNTERMEASURE ACTIONS.

A digital padlock is the central focus, rendered in a teal and blue color scheme. It features intricate circuit board patterns and glowing yellow and green dots. The background is a deep blue with floating binary digits (0s and 1s) and abstract light patterns. On the far left, there is a vertical orange bar with a series of white dots.

EXAMPLES

SCANNING A HOST



```
1 show("Digite o endereço de ip para verificar: ")
2 str:ip_address = input()
3 int:port_initial
4 int:port_final
5 int:port = 442
6 foreach port_initial = 442 to port_final = 445{
7     if scanhost(ip_address,port) == "open"{
8         show("Porta aberta: ")
9         show(port)
10    }else{
11        show("Porta fechada: ")
12        show(port)
13    }
14    port = port + 1
15 }
```

```
matheus1618@matheus1618-ThinkPad-E14:~/A
Digite o endereço de ip para verificar:
8.8.8.8
Porta fechada:
442
Porta aberta:
443
Porta fechada:
444
```


TRAFFIC ANALYSIS



```
1 int:initial
2 int:final
3 show("Starting Monitoration")
4 show("=====")
5 foreach initial = 1 to final = 10{
6     traffic_information()
7 }
8 show("=====")
9 show("Report complete")
```

```
matheus1618@matheus1618-ThinkPad-E14:~/
Starting Monitoration
=====

Timestamp: 2023-12-10 14:08:47
Packet Length: 275 bytes
Source MAC: 8c:c6:81:95:08:17
Destination MAC: a4:33:d7:c8:9b:33
Ethernet Type: 0x800
IPv4 Header:
Version: 4
IHL: 20 bytes
TTL: 64
Protocol: 6
Source IP: 192.168.15.67
Destination IP: 52.182.143.208

Timestamp: 2023-12-10 14:08:47
Packet Length: 172 bytes
Source MAC: 8c:c6:81:95:08:17
Destination MAC: a4:33:d7:c8:9b:33
Ethernet Type: 0x86dd
=====
Report complete
```

MALWARE DETECTION



```
1 str:file_name = "../LexParser/parser"
2 rule:rules_ryuk {
3   str:a = ".php?",
4   str:b = "uid=",
5   str:c = "&uname=",
6 }
7 if match(file_name, rules_ryuk){
8   show("Arquivo suspeito")
9 } else{
10  show("Arquivo normal")
11 }
```

```
matheus1618@matheus1618-ThinkPad-E14:~/D
Arquivo normal
```