SUBTERFUGE

MAN-IN-THE-MIDDLE MADE EASY

SUBTERFUGE

Christopher M. Shields
r00t0v3rr1d3

Matthew M. Toussain
0sm0s1z

BACKGROUND

Chris –

Custom Attack Tools and Project Management

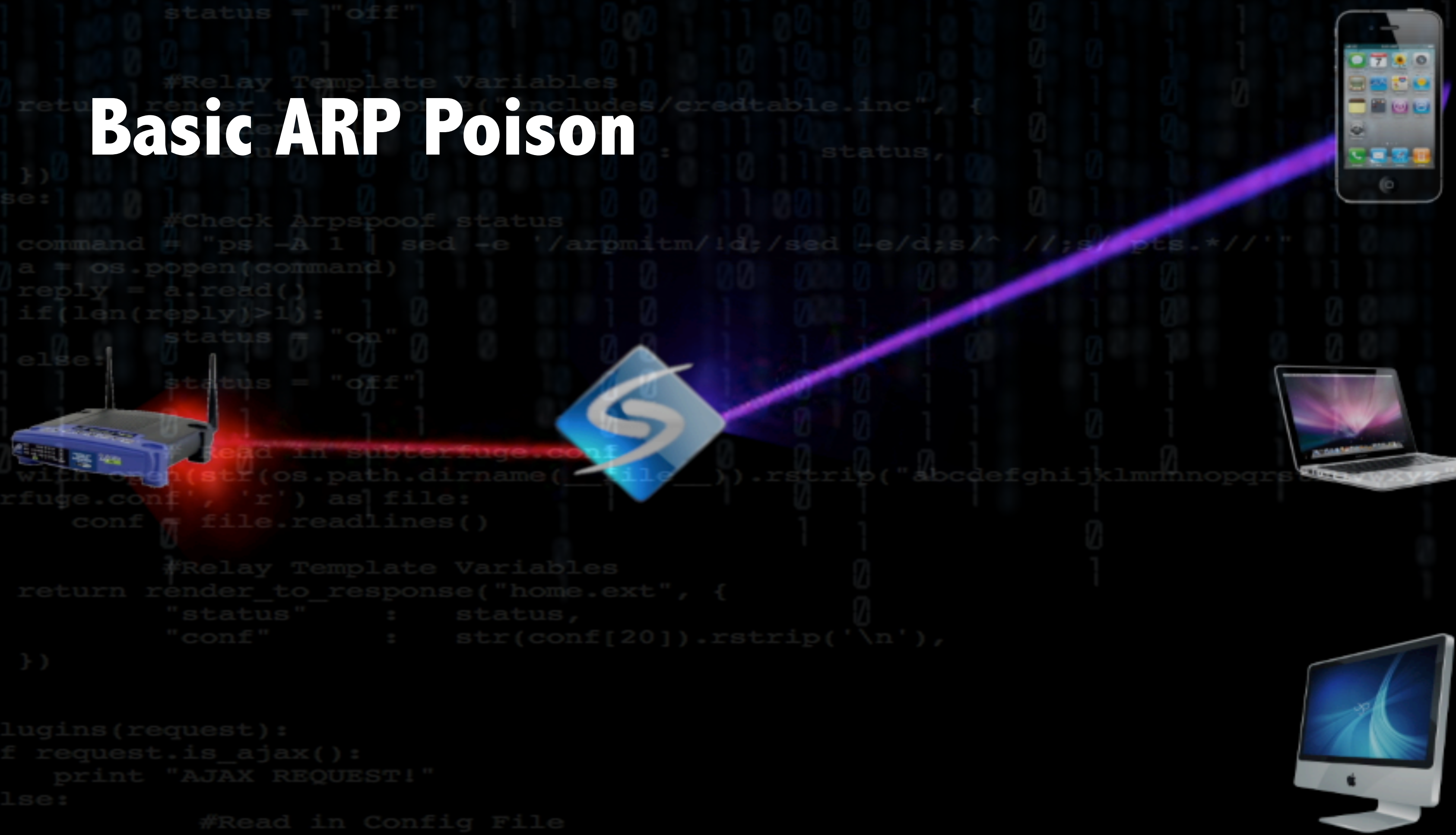
Matt –

Interface Design and Framework Development

SUBTERFUGE

ANATOMY OF THE ATTACK

Basic ARP Poison



ANATOMY OF THE ATTACK

ARPSPOOF

Heavy Network Traffic

Periods of MITM Loss

ARPMITM

Python Tool With Scapy

Intelligent Network Poison

Dynamic Poison Retention

ANATOMY OF THE ATTACK

SSLSTRIP

- HTTPS Downgrade Attack
- Use as a Web Proxy
- Customizations for Subterfuge

VECTOR TRANSPARENCY

TRADITIONAL SSL ATTACKS

CERTIFICATE ERRORS

EXPLOITING THE HUMAN

SUBTERFUGE

SUBTERFUGE

A New MITM Tool –

- Intuitive Interface
- Easy to use
- Silent and Stealthy
- Open Source

SUBTERFUGE

THE FRAMEWORK

- Server/Client Architecture
- MITM Utilities
- Module Builder
- Configuration Options

SUBTERFUGE

PLUGINS

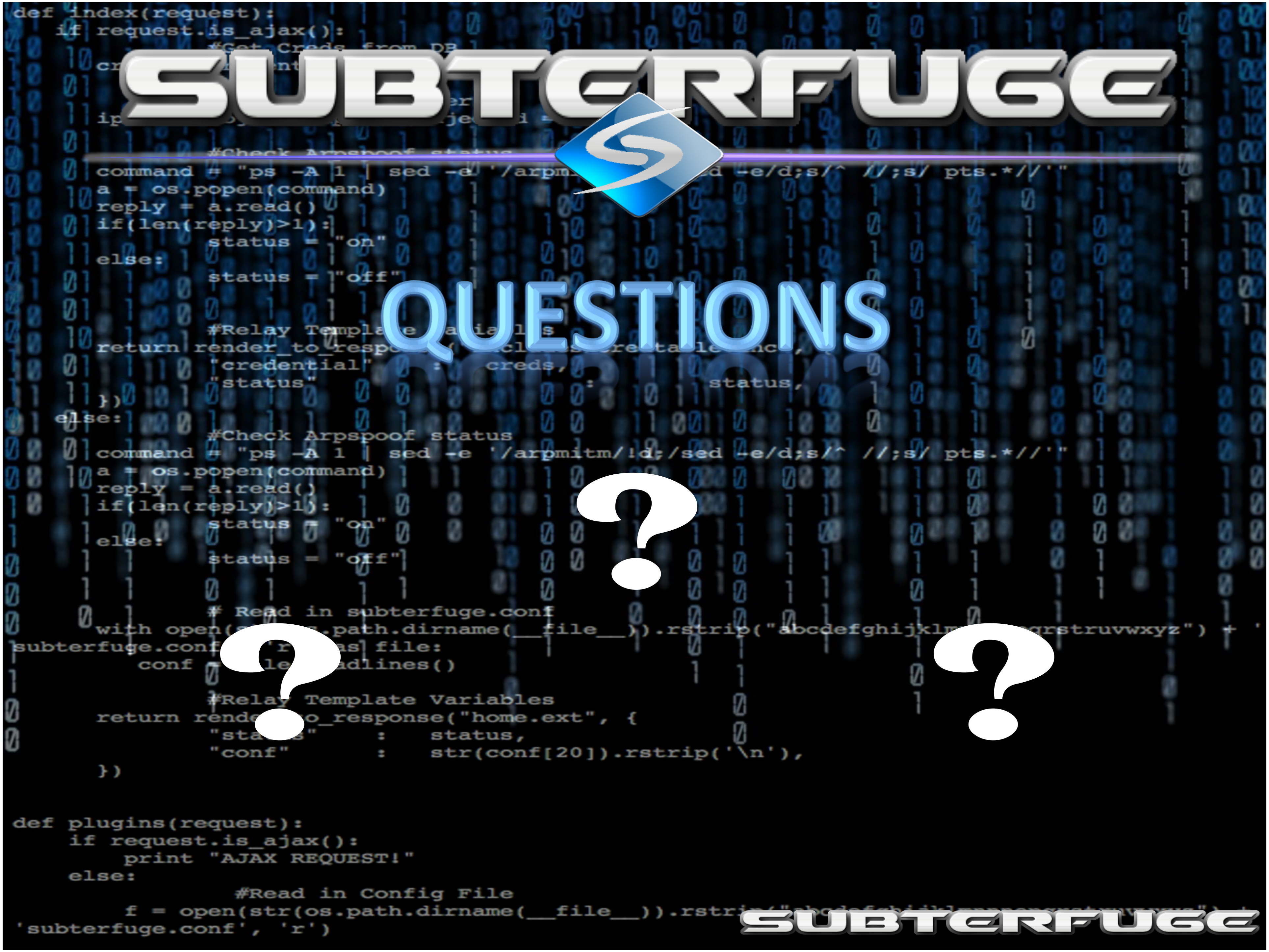
- ➢ Credential Harvesting
- ➢ HTTP Code Injection
- ➢ Denial of Service
- ➢ Network View

DEMONSTRATION

```
def index(request):
    if request.is_ajax():
                #Ge[illegible]Cr[illegible]omDB
        creds = cre[illegible]
    
            #Reset I[illegible]C[illegible]
        iptrack.objects.update(injected = "0[illegible]
    
                #Check Arpspoof status
        command = "ps -A 1 | sed -e '/arpm[illegible]d -e/d;s/^ //;s/ pts.*//'"
        a = os.popen(command)
        reply = a.read()
        if(len(reply)>1):
                status = "on"
        else:
                status = "off"
    
                #Relay Template Variables
        return render_to_response("includes/credtable.inc", {
                "credential"    :   creds,
                "status"                    :               status,
        })
    else:
                #Check Arpspoof status
        command = "ps -A 1 | sed -e '/arpmitm/!d;/sed -e/d;s/^ //;s/ pts.*//'"
        a = os.popen(command)
        reply = a.read()
        if(len(reply)>1):
                status = "on"
        else:
                status = "off"
    
                # Read in subterfuge.conf
        with open(str(os.path.dirname(__file__)).rstrip("abcdefghijklmnnopqrstruvwxyz") + '
subterfuge.conf', 'r') as file:
            conf = file.readlines()
    
                #Relay Template Variables
        return render_to_response("home.ext", {
                "status"    :   status,
                "conf"      :   str(conf[20]).rstrip('\n'),
        })


def plugins(request):
    if request.is_ajax():
        print "AJAX REQUEST!"
    else:
                  #Read in Config File
        f = open(str(os.path.dirname(__file__)).rstri[illegible] +
'subterfuge.conf', 'r')
```

SUBTERFUGE



QUESTIONS