

# 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

```
def index(request):
    if request.is_ajax():
        #Get Credentials from DB
        creds = credentials.objects.all()

        #Reset Injected output
        iptrack.objects.update(injected = "0")

        #Check Arspool 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
        subterfuge_conf = file.readlines(
            conf = file.readlines(
                #relay Template Variables
                use("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__)).rstrip("/") + "subterfuge.conf", 'r')
```

**SUBTERFUGE**

# ANATOMY OF THE ATTACK



## Basic ARP Poison



**SUBTERFUGE**

```
def index(request):
    if request.is_ajax():
        #Get arp from router
        creds = credntal.injected
        #Inject into router
        iptrack.objects.update(injected =
        #Check Arpspoof status
        command = "ps -A 1 | sed -e '/arpmitm/ld;/sed -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", {
            "status": status,
        })
    else:
        #Check Arpspoof status
        command = "ps -A 1 | sed -e '/arpmitm/ld;/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 file
        with open(str(os.path.dirname(__file__)).rstrip("abcdefghijklmnopqr
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__)).rstrip("abcdefghijklmnopqr
'subterfuge.conf', 'r')
```

# ANATOMY OF THE ATTACK



**ARPSPOOF**

**Heavy Network Traffic**

**Periods of MITM Loss**

**ARPMITM**

**Python Tool With Scapy**

**Intelligent Network Poison**

**Dynamic Poison**

**Retention**

**SUBTERFUGE**

```
def index(request):
    if request.is_ajax():
        #Get Credentials from ...
        creds = ...
        #Inject Credentials into ...
        iptrack.objects.update(injected = ...
        #Check Arpspoof status
        command = "ps -A | sed -e '/arpmitm/!d -e/d;s/^ //;s/ pts.*//'"
        a = os.popen(command)
        reply = a.read()
        if(len(reply)>1):
            status = "on"
        else:
            status = "off"
        return render_to_response("includes/credtable.inc", {
            "credential" : creds,
            "status" : status,
        })
    else:
        #Check Arpspoof status
        command = "ps -A | sed -e '/arpmitm/!d;/sed -e/d;s/^ //;s/ pts.*//'"
        #Check Arpspoof status
        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("abcdefhijklmnopqrstuvwxyz") + '
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__)).rstrip("abcdefhijklmnopqrstuvwxyz") + '
subterfuge.conf', 'r')
```

# ANATOMY OF THE ATTACK



## SSLSTRIP

➤ **HTTPS Downgrade Attack**

➤ **Use as a Web Proxy**

➤ **Customizations for Subterfuge**

```
def index(request):
    if request.is_ajax():
        #Get Credentials from Browser
        creds = request.all_headers['Cookie']
        #Inject Credentials into Browser
        iptrack.objects.update(injected =
        #Check Arnsboof status
        command = "ps -A | sed -e '/arpn...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 Arnsboof status
        command = "ps -A | sed -e '/arpn...d -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("abcdefghijklmnopqrstruvwxyz") +
        'subterfuge.conf', 'r') as file:
            conf = file.read()
        #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__)).rstrip("abcdefghijklmnopqrstruvwxyz") +
        'subterfuge.conf', 'r')
```

**SUBTERFUGE**

# VECTOR TRANSPARENCY



TRADITIONAL SSL ATTACKS

CERTIFICATE ERRORS

EXPLOITING THE HUMAN

```
def index(request):
    if request.is_ajax():
        #Read in subterfuge.conf
        creds = credentials.objects.all()
        #Read in object counter
        iptrack.objects.update(injected = 1)
        #Check Arpspoof status
        command = "ps -A | 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"
        #Relay Template Variables
        return render_to_response("includes/credtable.inc", {
            "credential" : creds,
            "status" : status,
        })
    else:
        #Check Arpspoof status
        command = "ps -A | 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("abcdefghijklmnopqrstruvwxyz") + '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__)).rstrip("abcdefghijklmnopqrstruvwxyz") + 'subterfuge.conf', 'r')
```

# SUBTERFUGE



## A New MITM Tool –

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

```
def index(request):
    if request.is_ajax():
        #Get Creds from DB
        cr = ...
        ip = ...
        #Check Arpspoof status
        command = "ps -A | sed -e '/arpmitm/!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 | sed -e '/arpmitm/!d;-e/d;s/^ //;/s/ pts.*//'"
        a = os.popen(command)
        reply = a.read()
        if(len(reply)>1):
            status = "on"
        else:
            status = "off"
        #Read in Config File
        with open(str(os.path.dirname(__file__)).rstrip("abcdefghijklmnopqrstruvwxyz") + '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__)).rstrip("abcdefghijklmnopqrstruvwxyz") + 'subterfuge.conf', 'r')
```

SUBTERFUGE

# SUBTERFUGE



## THE FRAMEWORK

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

```
def index(request):
    if request.is_ajax():
        #Get Creds from DB
        cr = ...
        ip = ...
        #Check Arpspoof status
        command = "ps -A | 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"
        #Relay Template Variables
        return render_to_response("includes/credtable.inc", {
            "credential" : creds,
            "status" : status,
        })
    else:
        #Check Arpspoof status
        command = "ps -A | 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('subterfuge.conf', 'r') as f:
            conf = f.rstrip("abcdefghijklmnopqrstuvwxyz") + '
subterfuge.conf'
            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__)).rstrip("/") + 'subterfuge.conf', 'r')
```

SUBTERFUGE



# SUBTERFUGE



## PLUGINS

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

SUBTERFUGE

# DEMONSTRATION



```
def index(request):
    if request.is_ajax():
        #Get creds from DB
        creds = credentials.objects.filter(a=1)
        #Reset injected counter
        iptrack.objects.update(injected = 0)
        #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"
        #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("abcdefghijklmnopqrstruvwxyz") +
            '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'),
        })

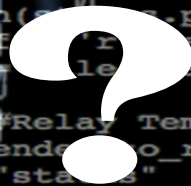
def plugins(request):
    if request.is_ajax():
        print "AJAX REQUEST!"
    else:
        #Read in Config File
        f = open(str(os.path.dirname(__file__)).rstrip("abcdefghijklmnopqrstruvwxyz") +
            'subterfuge.conf', 'r')
```

**SUBTERFUGE**

# SUBTERFUGE



## QUESTIONS



```
def index(request):
    if request.is_ajax():
        #Get Creds from DB
        cr = cursor.execute('select creds from creds')
        ip = request.GET.get('ip')
        #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"
        #Relay Template Variables
        return render_to_response('home.html', {
            '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(os.path.dirname(__file__)).rstrip("abcdefghijklmnopqrstuvwxyz") + '
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__)).rstrip("abcdefghijklmnopqrstuvwxyz") + '
subterfuge.conf', 'r')
```