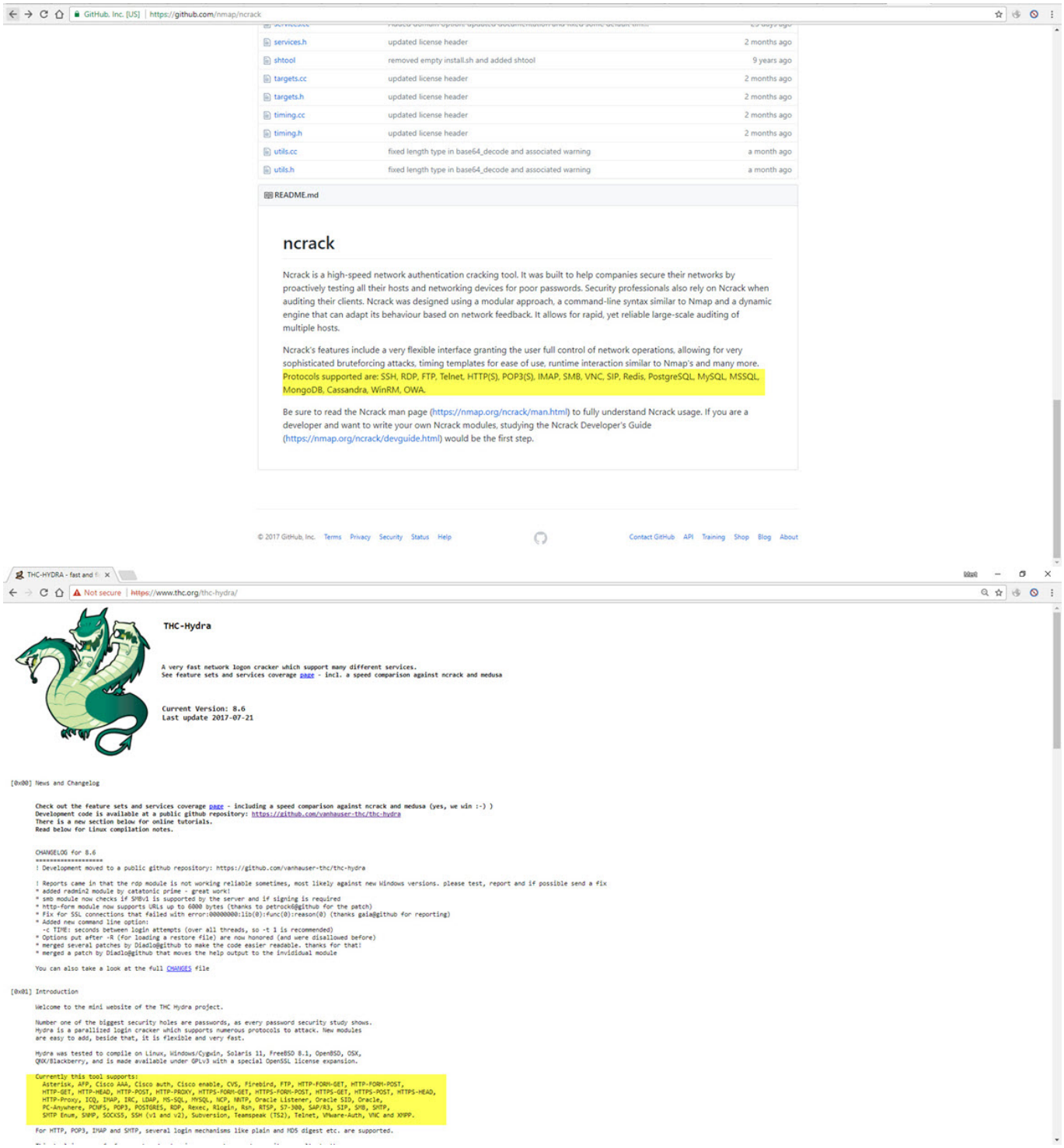
# Mini Tehdit İstihbaratı

written by Mert SARICA | 3 December 2018

Siber Güvenlik Merkezleri tarafından kurumlara gerçekleştirilen hedefli siber saldırıları tespit etmenin ve engellemenin öneminin yüksek olduğu günümüzde, basit bir kontrol ile tespit edilebilen düşük etkileşimli bir balküpü sistemi (Bkz: Balküpü Tespiti) maalesef kurumlar için kaynakları tüketen atıl bir sistem olmaktan öteye gidemiyor. Halbuki kolay bir şekilde kurup, yönetebileceğimiz, amacına hizmet edebilen bir balküpü sistemi yeri geldiğinde SIEM ile entegre edilerek kurumumuz için oldukça değerli bir mini siber tehdit istihbarat servisi olarak da kullanılabilir.

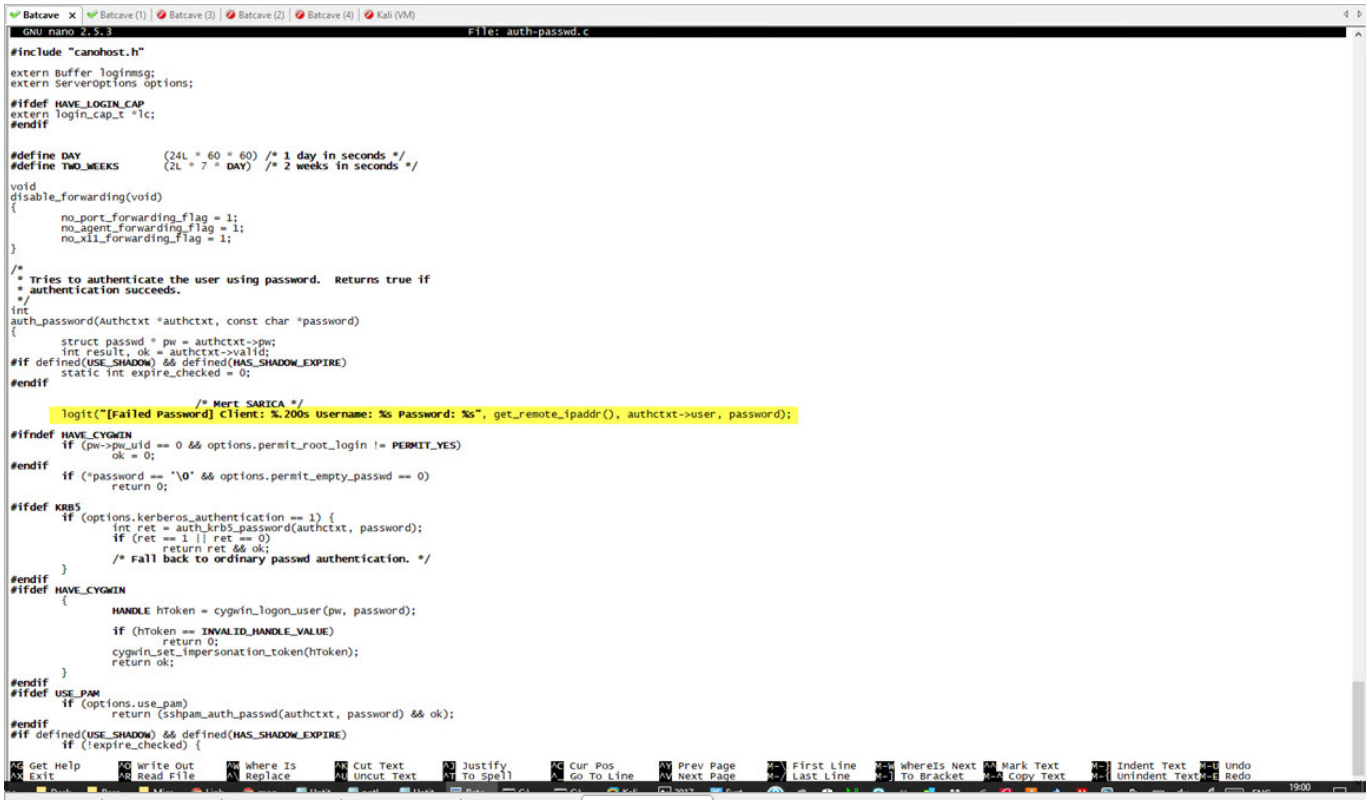
Bu ev yapımı, mini istihbarat servisi ile ayrıca kurumların hangi basit şifreler ile hangi servisler, protokoller üzerinden hacklendiğini, hangi ülkelerin siber saldırganlara ev sahipliği yaptığını bulmak da mümkün olabilirdi. Tuzak Sistem ile Hacker Avı çalışmamda olduğu gibi evimde konumlandıracağım bu sistemin hem basit bir şekilde siber saldırganlar tarafından tespit edilememesi hem de kısıtlı zamanımı çalmaması adına kolay yönetilebilmesi gereksinimlerimin başında geliyordu.

Bu konu üzerine yeteri kadar düşündükten ve taşındıktan sonra açık kaynak kodlu çeşitli servisleri modifiye edip bu servise yapılan hatalı giriş denemelerini (failed login attempt) istediğim şekilde kayıt altına alan bir sistem oluşturmaya karar verdim. Hangi servisler olacağına karar vermek için ise güvenlik uzmanları dışında bir o kadar da art niyetli kişiler tarafından da sıklıkla kullanılan ncrack ve THC-Hydra araçlarının destekledikleri protokollere göz atmaya karar verdim ve günün sonunda ssh, ftp, http, postgresql protokollerinde karar kıldım. Daha sonra Mini-PC üzerine kurduğum XenServer sanallaştırma sistemi üzerine iki tane sanal Ubuntu işletim sistemi kurdum. Ubuntulardan birine bu denemeleri kayıt altına alan loginmon isimli aracı bir diğerine loginmon aracının kayıtlarını (log) analiz etmek amacıyla Splunk Community sürümünü kurdum.




Tabii önümdeki en büyük engellerden biri bu protokolleri kullanan sunucu bileşenlerinin (sshd, postgresql, vsftpd gibi) hatalı giriş denemelerinde kullanılan parolaların açık (clear-text) olarak kayıt altına almalarıydı bu sebeple her bir sunucu bileşeninin kaynak koduna teker teker müdahale etmeye başladım.

Öncelikle openssh-7.2p2 paketini indirip auth-passwd.c dosyasını daha sonra postgres-10.0 paketini indirip src/backend/libpq/auth.c dosyasını ve ardından da vsftpd-3.0.3 paketini indirip prelogin.c ve logging.c dosyalarını, parolaların açık halini kayıt edecek şekilde değiştirdim ve derledim.

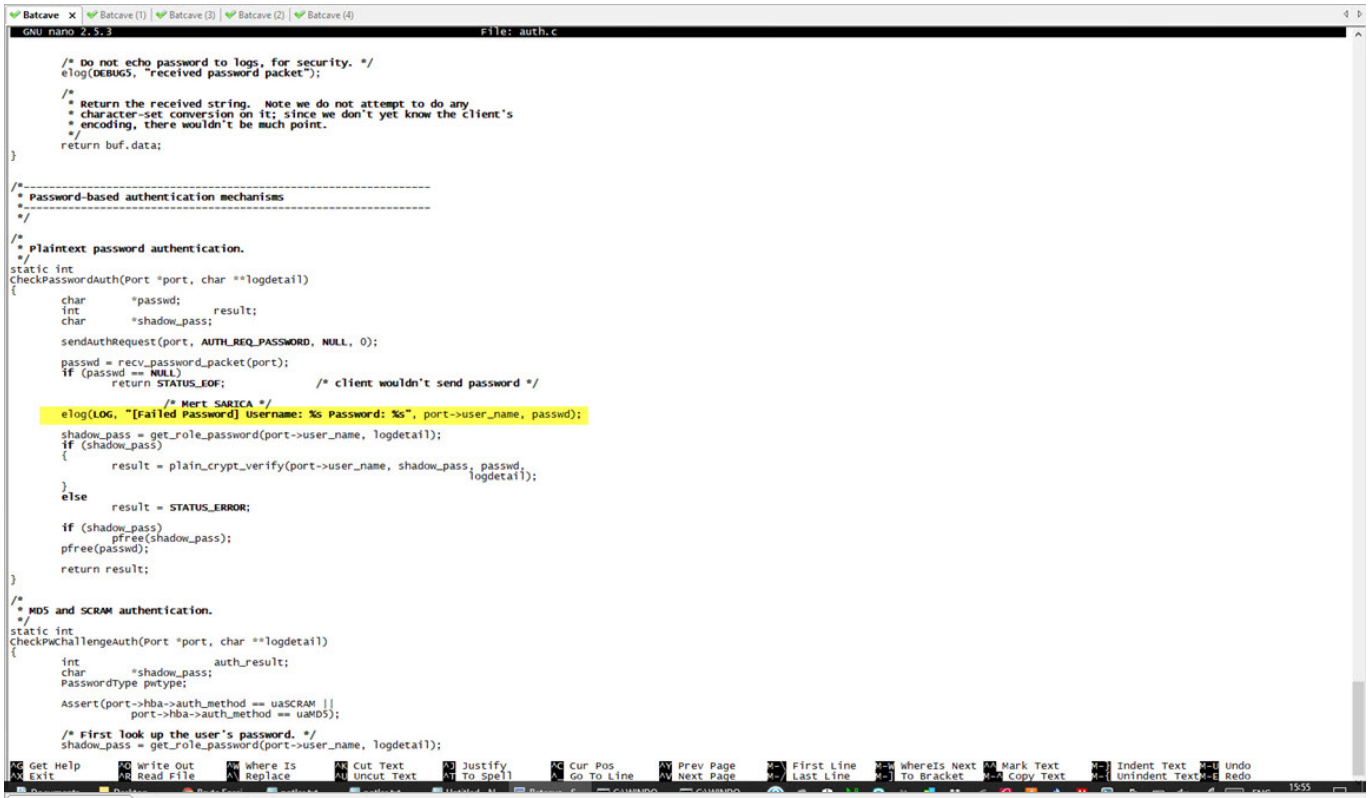



Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | **Kali (VM)** ×

```
root@kali:~# cat pass.txt
test
mert
dert
root@kali:~# hydra -t 32 -l root -P pass.txt 192.168.1.127 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-11-07 11:06:04
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 3 tasks per 1 server, overall 64 tasks, 3 login tries (l:1/p:3), ~0 tries per task
[DATA] attacking service ssh on port 22
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-11-07 11:06:05
root@kali:~#
```

Batcave | **Batcave (1)** × | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)

```
root@ubuntu:~# tail -n 9 /var/log/auth.log
Nov  7 19:06:05 ubuntu sshd[20604]: [Failed Password] Client: 192.168.1.144 Username: root Password: mert
Nov  7 19:06:05 ubuntu sshd[20603]: [Failed Password] Client: 192.168.1.144 Username: root Password: dert
Nov  7 19:06:05 ubuntu sshd[20602]: [Failed Password] Client: 192.168.1.144 Username: root Password: test
Nov  7 19:06:05 ubuntu sshd[20604]: Failed password for root from 192.168.1.144 port 64781 ssh2
Nov  7 19:06:05 ubuntu sshd[20603]: Failed password for root from 192.168.1.144 port 64780 ssh2
Nov  7 19:06:05 ubuntu sshd[20602]: Failed password for root from 192.168.1.144 port 64779 ssh2
Nov  7 19:06:05 ubuntu sshd[20604]: Connection closed by 192.168.1.144 port 64781 [preauth]
Nov  7 19:06:05 ubuntu sshd[20603]: Connection closed by 192.168.1.144 port 64780 [preauth]
Nov  7 19:06:05 ubuntu sshd[20602]: Connection closed by 192.168.1.144 port 64779 [preauth]
root@ubuntu:~#
```



```
Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4)
 GNU nano 2.5.3                                File: auth.c

        /* Do not echo password to logs, for security. */
        elog(DEBUG5, "received password packet");

        /*
         * Return the received string.  Note we do not attempt to do any
         * character-set conversion on it; since we don't yet know the client's
         * encoding, there wouldn't be much point.
         */
        return buf.data;
}


/*----------------------------------------------------------------
 * Password-based authentication mechanisms
 *----------------------------------------------------------------
 */

/*
 * Plaintext password authentication.
 */
static int
CheckPasswordAuth(Port *port, char **logdetail)
{
        char       *passwd;
        int                result;
        char       *shadow_pass;

        sendAuthRequest(port, AUTH_REQ_PASSWORD, NULL, 0);

        passwd = recv_password_packet(port);
        if (passwd == NULL)
                return STATUS_EOF;              /* client wouldn't send password */

                        /* Mert SARICA */
      elog(LOG, "[Failed Password] Username: %s Password: %s", port->user_name, passwd);

        shadow_pass = get_role_password(port->user_name, logdetail);
        if (shadow_pass)
        {
                result = plain_crypt_verify(port->user_name, shadow_pass, passwd,
                                                                        logdetail);
        }
        else
                result = STATUS_ERROR;

        if (shadow_pass)
                pfree(shadow_pass);
        pfree(passwd);

        return result;
}

/*
 * MD5 and SCRAM authentication.
 */
static int
CheckPWChallengeAuth(Port *port, char **logdetail)
{
        int                auth_result;
        char       *shadow_pass;
        PasswordType pwtype;

        Assert(port->hba->auth_method == uaSCRAM ||
                   port->hba->auth_method == uaMD5);

        /* First look up the user's password. */
        shadow_pass = get_role_password(port->user_name, logdetail);

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page  ^V First Line  M-W WhereIs Next  M-A Mark Text  M-] Indent Text  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page  M-/ Last Line  M-] To Bracket  M-6 Copy Text  M-[ Unindent Text M-E Redo
```

Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4)

```
root@ubuntu:~/honeypot# hydra -l root -P pass.txt 127.0.0.1 postgres
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-11-05 15:56:53
[DATA] max 3 tasks per 1 server, overall 3 tasks, 3 login tries (l:1/p:3), ~1 try per task
[DATA] attacking postgres://127.0.0.1:5432/
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-11-05 15:56:54
root@ubuntu:~/honeypot# cat /var/log/pgsql
2017-11-05 15:56:27.206 +03 [10974] LOG:  listening on IPv6 address "::1", port 5432
2017-11-05 15:56:27.206 +03 [10974] LOG:  listening on IPv4 address "127.0.0.1", port 5432
2017-11-05 15:56:27.208 +03 [10974] LOG:  listening on Unix socket "/tmp/.s.PGSQL.5432"
2017-11-05 15:56:27.228 +03 [10975] LOG:  database system was shut down at 2017-11-05 15:54:28 +03
2017-11-05 15:56:27.231 +03 [10974] LOG:  database system is ready to accept connections
2017-11-05 15:56:54.091 +03 [10991] LOG:  [Failed Password] Username: root Password: dert
2017-11-05 15:56:54.091 +03 [10991] FATAL:  password authentication failed for user "root"
2017-11-05 15:56:54.091 +03 [10991] DETAIL:  Role "root" does not exist.
        Connection matched pg_hba.conf line 86: "host    all             all             0.0.0.0/0               password"
2017-11-05 15:56:54.094 +03 [10990] LOG:  incomplete startup packet
2017-11-05 15:56:54.095 +03 [10992] LOG:  [Failed Password] Username: root Password: mert
2017-11-05 15:56:54.095 +03 [10992] FATAL:  password authentication failed for user "root"
2017-11-05 15:56:54.095 +03 [10992] DETAIL:  Role "root" does not exist.
        Connection matched pg_hba.conf line 86: "host    all             all             0.0.0.0/0               password"
2017-11-05 15:56:54.096 +03 [10989] LOG:  incomplete startup packet
2017-11-05 15:56:54.099 +03 [10994] LOG:  [Failed Password] Username: root Password: test
2017-11-05 15:56:54.099 +03 [10994] FATAL:  password authentication failed for user "root"
2017-11-05 15:56:54.099 +03 [10994] DETAIL:  Role "root" does not exist.
        Connection matched pg_hba.conf line 86: "host    all             all             0.0.0.0/0               password"
2017-11-05 15:56:54.103 +03 [10993] LOG:  incomplete startup packet
root@ubuntu:~/honeypot#
```

Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)

GNU nano 2.5.3 File: /root/honeypot/vsftpd-3.0.3/prelogin.c

```
  if (tunable_userlist_enable)
  {
    int located = str_contains_line(&p_sess->userlist_str, &p_sess->user_str);
    if ((located && tunable_userlist_deny) ||
        (!located && !tunable_userlist_deny))
    {
      check_login_delay();
      vsf_cmdio_write(p_sess, FTP_LOGINERR, "Permission denied.");
      check_login_fails(p_sess);
      str_empty(&p_sess->user_str);
      return;
    }
  }
  if (is_anon && tunable_no_anon_password)
  {
    /* Fake a password */
    str_alloc_text(&p_sess->ftp_arg_str, "<no password>");
    handle_pass_command(p_sess);
  }
  else
  {
    vsf_cmdio_write(p_sess, FTP_GIVEPWORD, "Please specify the password.");
  }
}

static void
handle_pass_command(struct vsf_session* p_sess)
{
  if (str_isempty(&p_sess->user_str))
  {
    vsf_cmdio_write(p_sess, FTP_NEEDUSER, "Login with USER first.");
    return;
  }

              /* Mert SARICA */
  struct mystr str_log_line = INIT_MYSTR;
  str_alloc_text(&str_log_line, "[Failed Password] Client: ");
  str_append_str(&str_log_line, &p_sess->remote_ip_str);
  str_append_text(&str_log_line, "  Username: ");
  str_append_str(&str_log_line, &p_sess->user_str);
  str_append_text(&str_log_line, "  Password: ");
  str_append_str(&str_log_line, &p_sess->ftp_arg_str);
  vsf_log_line(p_sess, kVSFLogEntryConnection, &str_log_line);

  /* These login calls never return if successful */
  if (tunable_one_process_model)
  {
    vsf_one_process_login(p_sess, &p_sess->ftp_arg_str);
  }
  else
  {
    vsf_two_process_login(p_sess, &p_sess->ftp_arg_str);
  }
  vsf_cmdio_write(p_sess, FTP_LOGINERR, "Login incorrect.");
  check_login_fails(p_sess);
  str_empty(&p_sess->user_str);
  /* FALLTHRU if login fails */
}

static void check_login_delay()
{
  if (tunable_delay_failed_login)
  {
    vsf_sysutil_sleep((double) tunable_delay_failed_login);
  }
}

static void check_login_fails(struct vsf_session* p_sess)
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page ^_ First Line M-W WhereIs Next M-A Mark Text M-] Indent Text M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page M-/ Last Line M-] To Bracket M-^ Copy Text M-[ Unindent Text M-E Redo

17:54

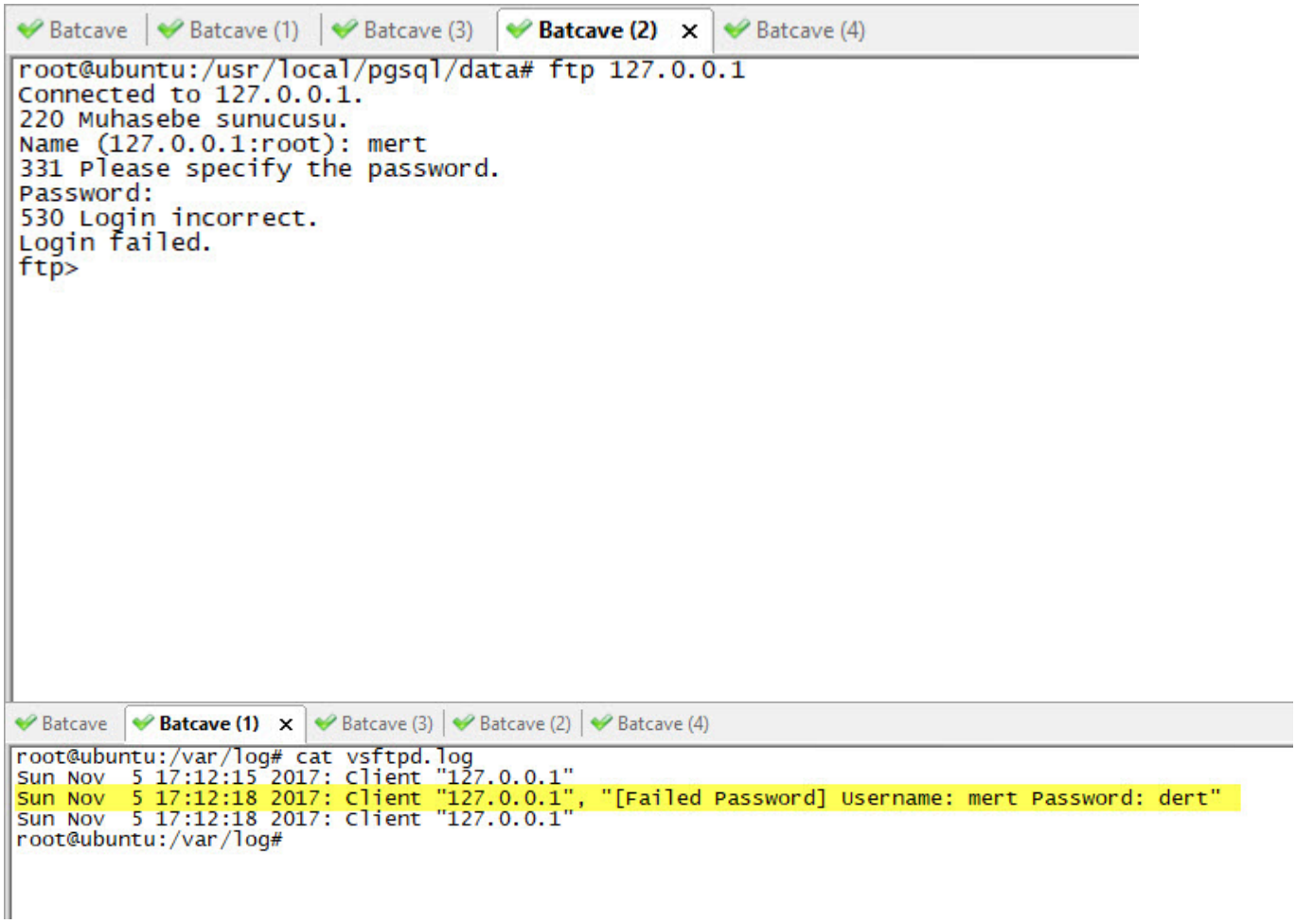
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)

GNU nano 2.5.3 File: /root/honeypot/vsftpd-3.0.3/logging.c

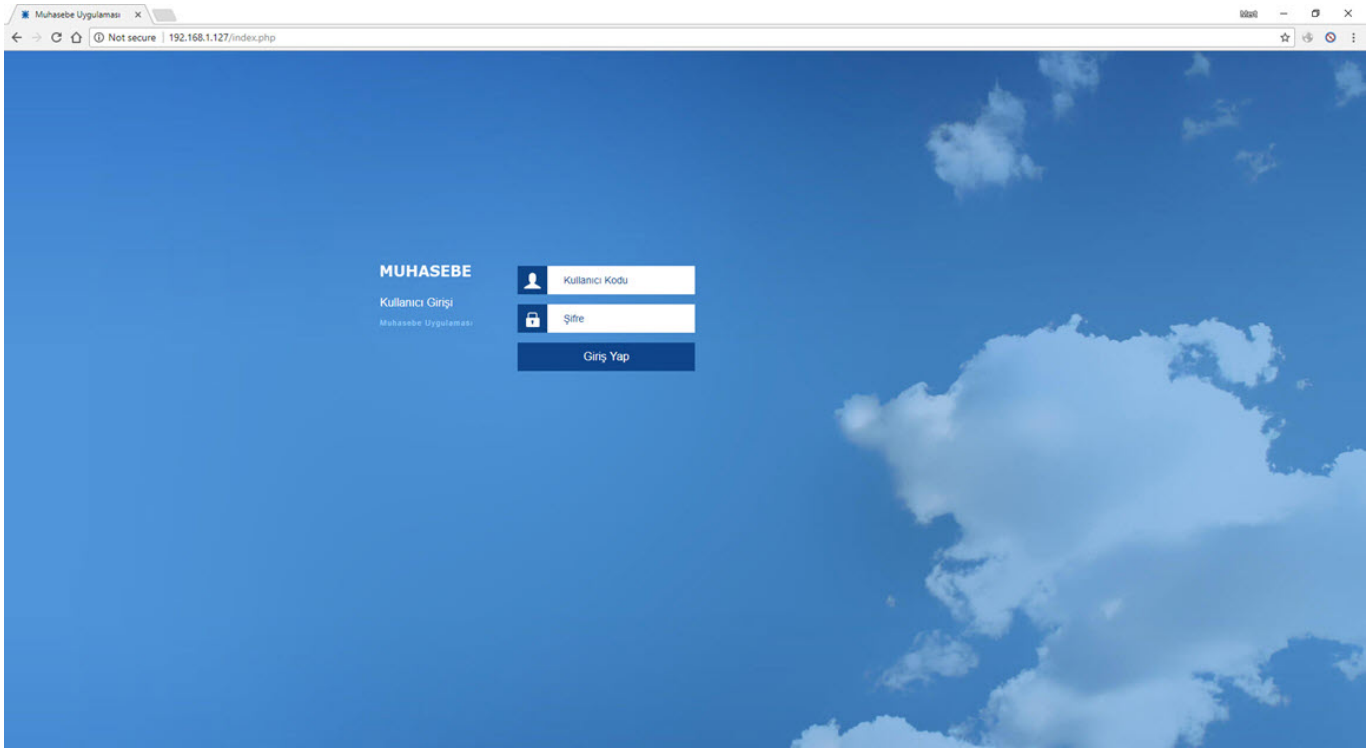
```
    str_append_text(p_str, "o ");
  }
  /* Access mode: anonymous/real user, and identity */
  if (p_sess->is_anonymous && !p_sess->is_guest)
  {
    str_append_text(p_str, "a ");
    str_append_str(p_str, &p_sess->anon_pass_str);
  }
  else
  {
    if (p_sess->is_guest)
    {
      str_append_text(p_str, "g ");
    }
    else
    {
      str_append_text(p_str, "r ");
    }
    str_append_str(p_str, &p_sess->user_str);
  }
  str_append_char(p_str, ' ');
  /* Service name, authentication method, authentication user id */
  str_append_text(p_str, "ftp 0 * ");
  /* Completion status */
  if (succeeded)
  {
    str_append_char(p_str, 'c');
  }
  else
  {
    str_append_char(p_str, 'i');
  }
}

            /* Mert SARICA */
static void
vsf_log_do_log_vsftpd_format(struct vsf_session* p_sess, struct mystr* p_str,
                             int succeeded, enum EVSFLogEntryType what,
                             const struct mystr* p_log_str)
{
  str_empty(p_str);
  if (!tunable_syslog_enable)
  {
    /* Date - vsf_sysutil_get_current_date updates cached time */
    str_append_text(p_str, vsf_sysutil_get_current_date());
  }
  if (!str_isempty(p_log_str))
  {
    str_append_text(p_str, ", \"");
    str_append_str(p_str, p_log_str);
    str_append_char(p_str, '"');
  }
}
```

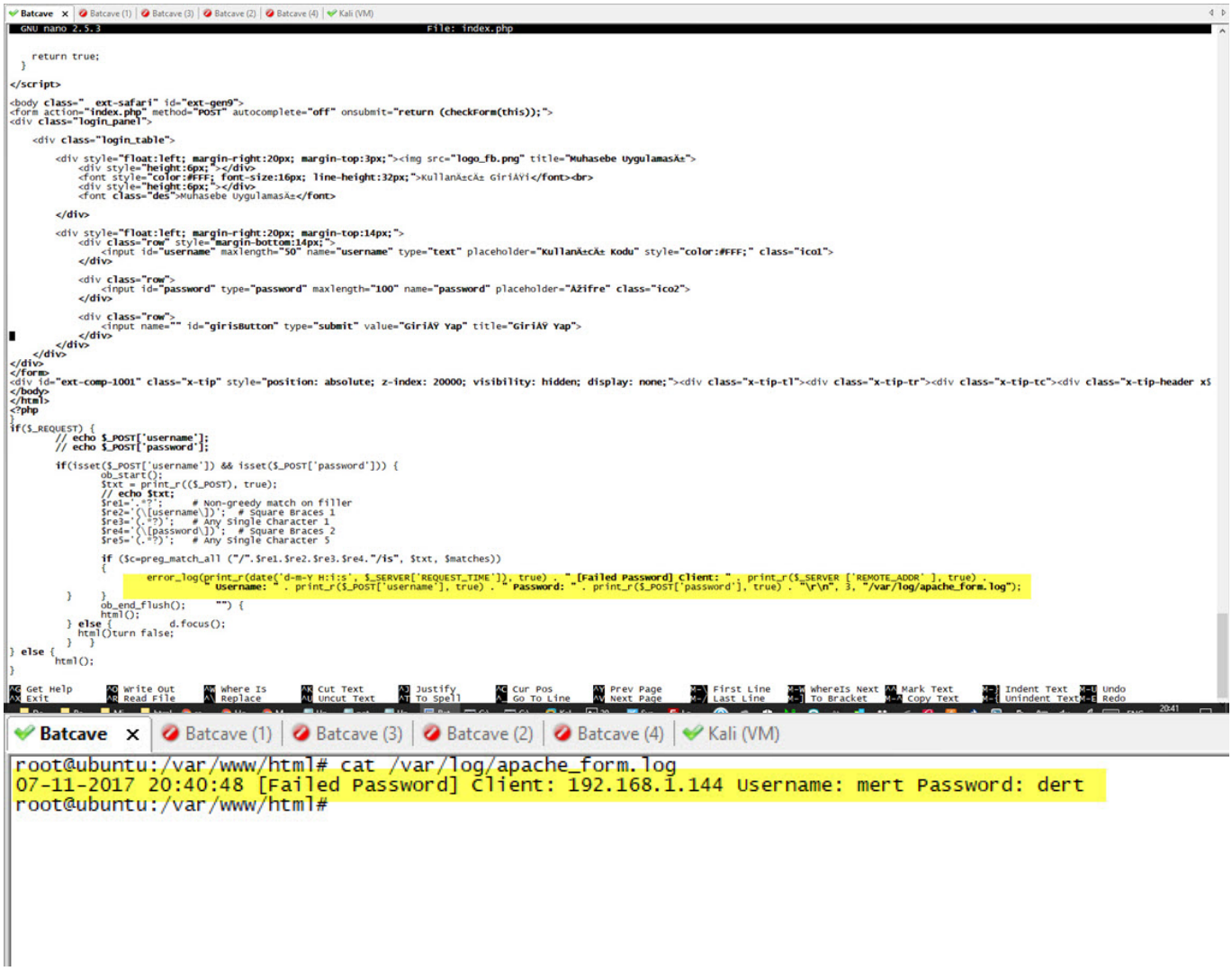
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page ^_ First Line M-W WhereIs Next M-A Mark Text M-] Indent Text M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page M-/ Last Line M-] To Bracket M-^ Copy Text M-[ Unindent Text M-E Redo

17:54




Son olarak ise web sunucusunun ana klasöründe oluşturduğum index.php dosyasının, yapılan hatalı giriş denemelerini kayıt altına almasını sağladım.






Python ile geliştirdiğim loginmon isimli aracı da saat başı /var/log klasörü altında yer alan ve açık halde saklanan parolaları, hatalı giriş denemesi yapan kullanıcı adları, ip adresleri ve tarihler ile alıp, Splunk'a gönderecek şekilde hazırladım.

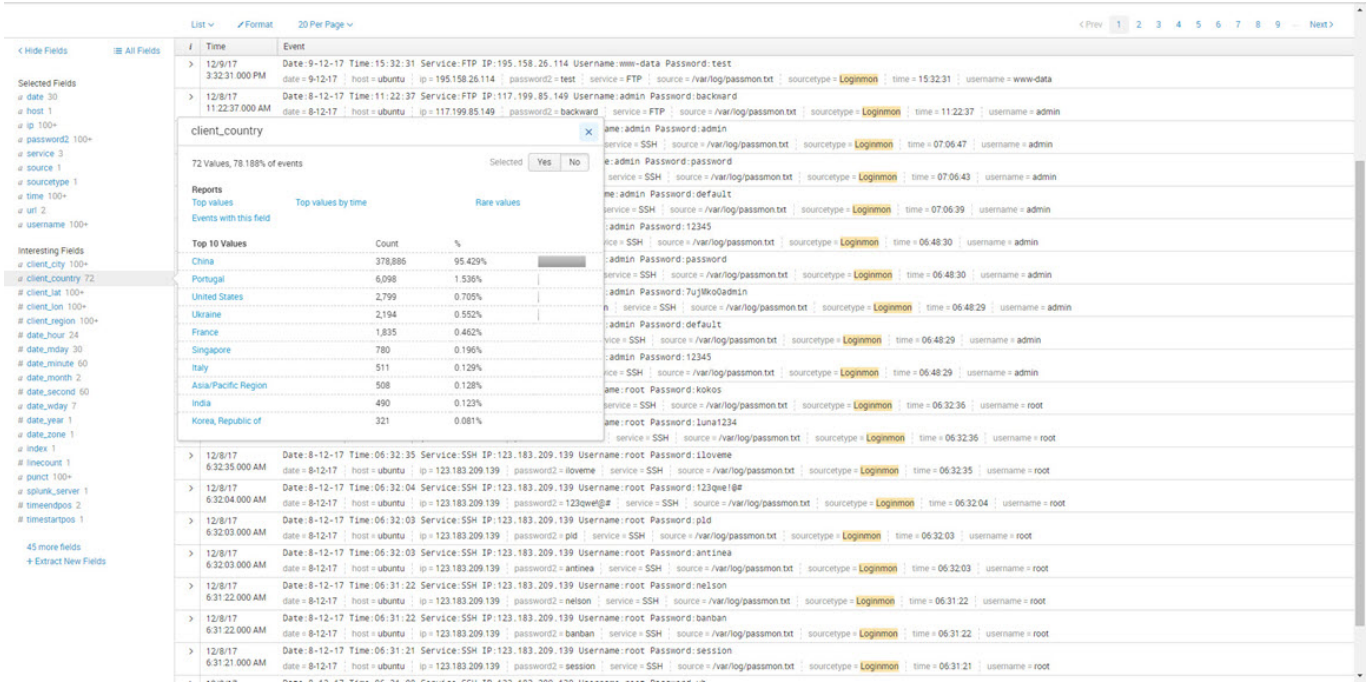
```
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)
  GNU nano 2.5.3                          File: /etc/cron.hourly/loginmon

#!/usr/bin/python
# Failed Login Attempt Monitor v1.0
# Author: Mert SARICA
# E-mail: mert [ . ] sarica [ @ ] gmail [ . ] com
# URL: https://www.mertsarica.com

import time
import os
import datetime
import hashlib
import sys
import re
import smtplib
from email.header import Header
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders

debug = 0

# Log Files
logfile = "/var/log/passmon.txt"
ftplog = "/var/log/vsftpd.log"
authlog = "/var/log/auth.log"
pglog = "/var/log/pgsql.log"
htalog = "/var/log/htpasswd.log"
formlog = "/var/log/apache_form.log"

# E-mail Parameters
sender = "mert.sarica@gmail.com"
recipient = "mert.sarica@gmail.com"
server = "127.0.0.1"
port = 25

def send_data():
        try:
                FILE  = open (logfile,"rb" )
        except Exception,e:
                if debug:
                        print "[*] send_data error: " + str(e)
                return

        subject = "Failed Login Attempts"

        msg = MIMEMultipart()
        msg.set_charset("iso-8859-9")

        msg['Subject'] = str(Header(subject, "iso-8859-9"))
        msg['From'] = sender
        msg['To'] = recipient
        part = MIMEBase('application', "octet-stream")
        part.set_payload(FILE.read())
        encoders.encode_base64(part)
        part.add_header('Content-Disposition', 'attachment; filename="passmon.txt"')

        # Attach parts into message container.
        # According to RFC 2046, the last part of a multipart message, in this case
        # the HTML message, is best and preferred.
        msg.attach(part)

        try:
                session = smtplib.SMTP(server, port)
                session.ehlo()
                session.sendmail(sender, recipient, msg.as_string())
                                                [ Read 366 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page  M-\ First Line  M-W WhereIs Next  M-A Mark Text  M-] Indent Text  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Linter ^_ Go To Line ^V Next Page  M-/ Last Line  M-] To Bracket  M-6 Copy Text  M-[ Unindent Text M-E Redo
```

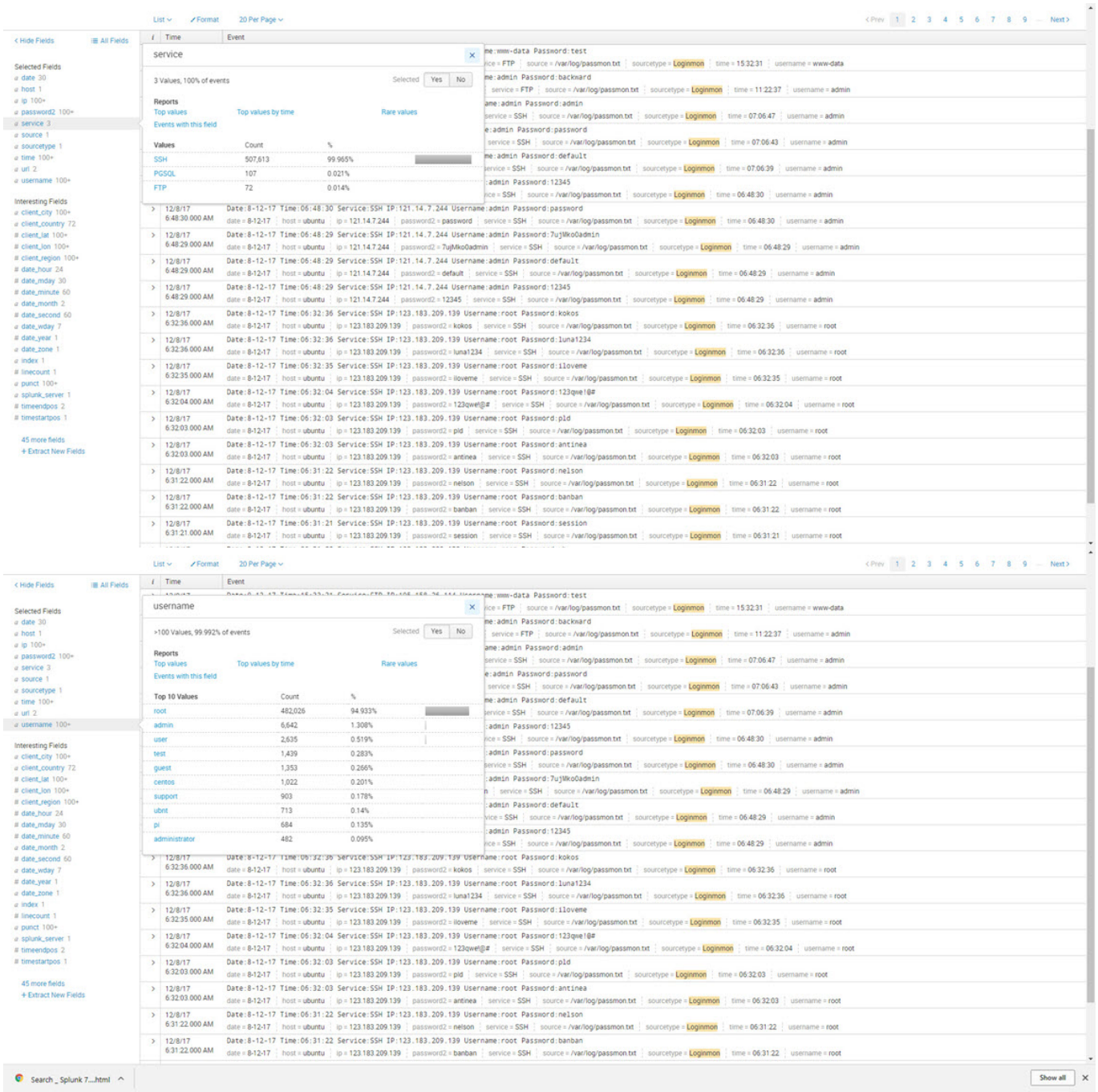
```
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)
  GNU nano 2.5.3                          File: /etc/cron.hourly/loginmon


        rg = re.compile(re1+re2+re3+re4+re5+re6+re7+re8+re9+re10+re11+re12,re.IGNORECASE|re.MULTILINE)

        for m in rg.finditer(txt):
                if m:
                        date1=m.group(1)
                        mydate = datetime.datetime.strptime(date1, '%d-%m-%Y')
                        time1=m.group(2)
                        sbraces1=m.group(3)
                        var1=m.group(4)
                        ipaddress1=m.group(5)
                        word1=m.group(6)
                        var2=m.group(7)
                        word2=m.group(8)
                        var3=m.group(9)
                        line = "Date:" + mydate.strftime('%-d %b') + " Time:" + time1 + " Service:WEBFORM IP:" + ipaddress1 + " Username:" + var2 + " Password:" + var3
                        print "\t[+] " + line
                        log(line)


def banner():
        os.system("clear")
        print "_____________________________________________________"
        print u"Failed Login Attempt Monitor - https://www.mertsarica.com"
        print "_____________________________________________________"


def main():
#       while (1==1):
                try:
                        print "[*] Analyzing logs..."
                        parse_authlog()
                        parse_ftplog()
                        parse_pglog()
                        parse_htalog()
                        parse_formlog()


                except KeyboardInterrupt:
                        banner()
                        print "[*] Bye..."
                        sys.exit(1)
                except Exception, e:
                        print "[*] Error:", str(e)
                        sys.exit(1)

if __name__ == '__main__':
        banner()
        main()

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page  M-\ First Line  M-W WhereIs Next  M-A Mark Text  M-] Indent Text  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Linter ^_ Go To Line ^V Next Page  M-/ Last Line  M-] To Bracket  M-6 Copy Text  M-[ Unindent Text M-E Redo
```

```
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)
_____________________________________________________
Failed Login Attempt Monitor - https://www.mertsarica.com
_____________________________________________________
[*] Analyzing logs...
        [+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:dert
        [+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:mert
        [+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:test
        [+] Date:5 Nov Time:17:53:23 Service:FTP IP:192.168.1.144 Username:mert Password:dert
        [+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:mert
        [+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:test
        [+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:dert
        [+] Date:8 Nov Time:20:16:10 Service:HTACCESS IP:192.168.1.144 Username:mert Password:dert URL:/admin/
        [+] Date:8 Nov Time:20:16:12 Service:HTACCESS IP:192.168.1.144 Username:mert Password:dert URL:/admin/
        [+] Date:7 Nov Time:20:40:48 Service:WEBFORM IP:192.168.1.144 Username:mert Password:dert
```

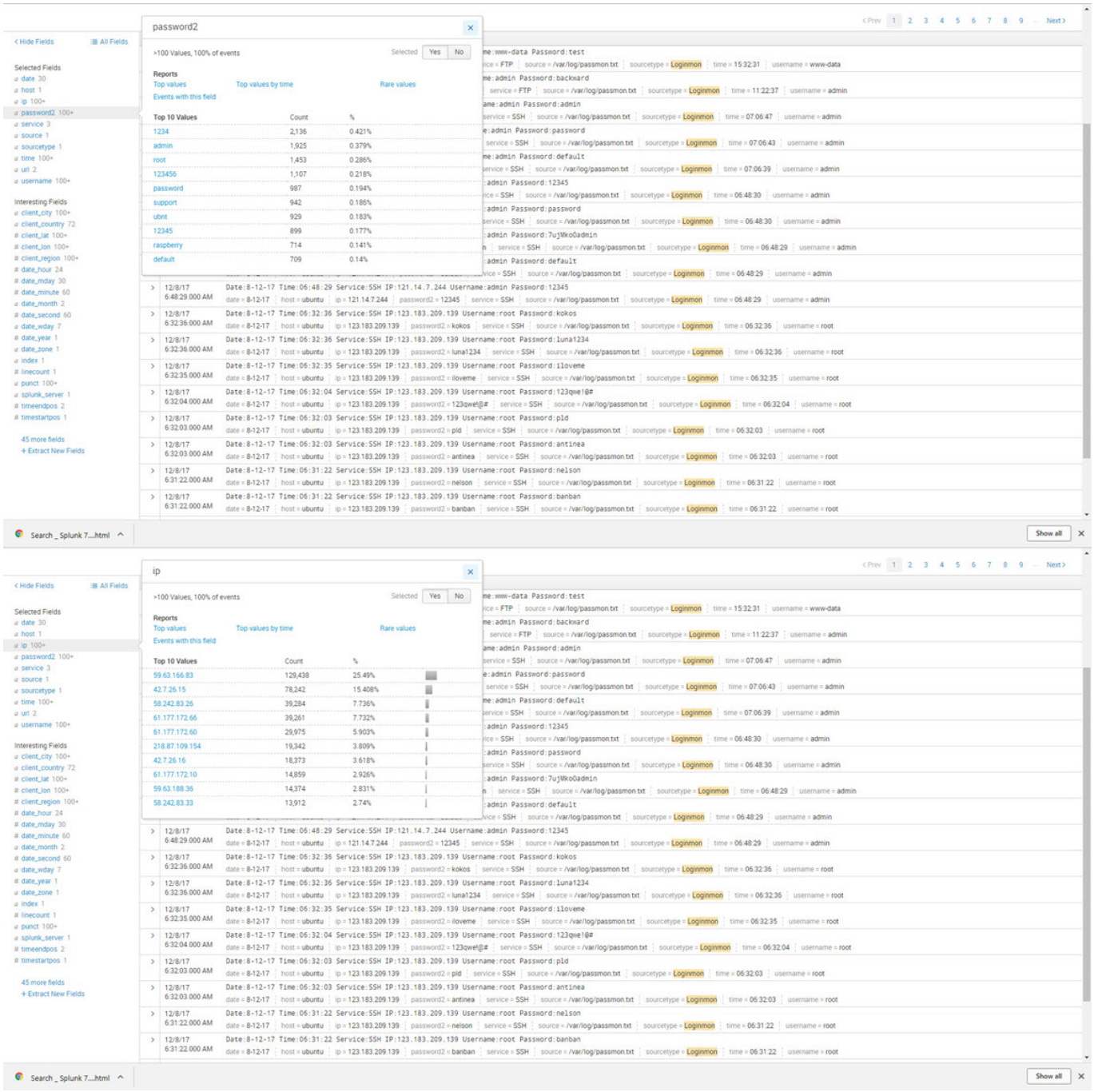
```
Batcave
root@ubuntu:/var/log# head -n 15 passmon.txt
Date:4-12-17 Time:17:18:10 Service:SSH IP:59.63.166.83 Username:root Password:1.0
Date:4-12-17 Time:16:35:44 Service:SSH IP:59.63.166.83 Username:root Password:root!@#
Date:5-12-17 Time:19:16:21 Service:SSH IP:58.242.83.25 Username:root Password:jumpstart
Date:4-12-17 Time:16:35:48 Service:SSH IP:59.63.166.83 Username:root Password:root!@#
Date:4-12-17 Time:16:35:49 Service:SSH IP:59.63.166.83 Username:root Password:pumpkin
Date:5-12-17 Time:19:16:22 Service:SSH IP:58.242.83.25 Username:root Password:jun123
Date:6-12-17 Time:09:23:45 Service:SSH IP:42.7.26.16 Username:root Password:qweasdqwe
Date:4-12-17 Time:16:35:50 Service:SSH IP:59.63.166.83 Username:root Password:planet1
Date:6-12-17 Time:09:23:46 Service:SSH IP:42.7.26.16 Username:root Password:qweasdzxc!@#
Date:5-12-17 Time:19:16:23 Service:SSH IP:58.242.83.25 Username:root Password:jussi
Date:4-12-17 Time:16:35:50 Service:SSH IP:59.63.166.83 Username:root Password:powerpc
Date:6-12-17 Time:09:23:47 Service:SSH IP:42.7.26.16 Username:root Password:qazwsxedc!@#123
Date:5-12-17 Time:19:16:24 Service:SSH IP:58.242.83.25 Username:root Password:justdoit
Date:6-12-17 Time:09:23:48 Service:SSH IP:42.7.26.16 Username:root Password:!@#qwe!@#
Date:5-12-17 Time:19:16:25 Service:SSH IP:58.242.83.25 Username:root Password:kaiser
root@ubuntu:/var/log#

root@ubuntu:~# /opt/splunkforwarder/bin/splunk add monitor /var/log/passmon.txt -sourcetype Loginmon -index main
Added monitor of '/var/log/passmon.txt'.
root@ubuntu:~#
```

Yaklaşık 1 ay sonunda Splunk üzerinde biriken ~500.000 kayıt üzerinde gerçekleştirmiş olduğum analizde, sistemime gerçekleştirilen deneme yanılma ve sözlük saldırılarının en çok Çin'den gerçekleştirildiğini, en fazla saldırı alan servisin SSHD olduğunu, kullanıcı adı olarak en çok root, parola için ise 1234 ile deneme yapıldığını öğrenmiş oldum.




List
Format
20 Per Page
Prev 1 2 3 4 5 6 7 8 9 Next
Hide Fields
All Fields
Time
Event
service
3 Values, 100% of events
Selected Yes No
Reports
Top values
Top values by time
Rare values
Events with this field
Values Count %
SSH 507,613 99.965%
PGSQL 107 0.021%
FTP 72 0.014%
Selected Fields
a date 30
a host 1
a ip 100+
a password2 100+
a service 3
a source 1
a sourcetype 1
a time 100+
a url 2
a username 100+
Interesting Fields
a client_city 100+
a client_country 72
# client_lat 100+
# client_lon 100+
# client_region 100+
# date_hour 24
# date_mday 30
# date_minute 60
a date_month 2
# date_second 60
a date_wday 7
# date_year 1
a date_zone 1
a index 1
# linecount 1
a punct 100+
a splunk_server 1
# timeendpos 2
# timestartpos 1
45 more fields
+ Extract New Fields


List
Format
20 Per Page
Prev 1 2 3 4 5 6 7 8 9 Next
Hide Fields
All Fields
Time
Event
username
>100 Values, 99.992% of events
Selected Yes No
Reports
Top values
Top values by time
Rare values
Events with this field
Top 10 Values Count %
root 482,026 94.933%
admin 6,642 1.308%
user 2,635 0.519%
test 1,439 0.283%
guest 1,353 0.266%
centos 1,022 0.201%
support 903 0.178%
ubnt 713 0.14%
pi 684 0.135%
administrator 482 0.095%
Selected Fields
a date 30
a host 1
a ip 100+
a password2 100+
a service 3
a source 1
a sourcetype 1
a time 100+
a url 2
a username 100+
Interesting Fields
a client_city 100+
a client_country 72
# client_lat 100+
# client_lon 100+
# client_region 100+
# date_hour 24
# date_mday 30
# date_minute 60
a date_month 2
# date_second 60
a date_wday 7
# date_year 1
a date_zone 1
a index 1
# linecount 1
a punct 100+
a splunk_server 1
# timeendpos 2
# timestartpos 1
45 more fields
+ Extract New Fields
Search _ Splunk 7....html
Show all







Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.