John Menerick

Open Source Fairy Dust
SECURITY DRAGON @ NETSUITE

THE VIEWS AND OPINIONS EXPRESSED HERE ARE MY OWN ONLY AND IN NO WAY REPRESENT THE VIEWS, POSITIONS OR OPINIONS - EXPRESSED OR IMPLIED - OF MY EMPLOYER (PRESENT AND PAST) OR ANYONE ELSE. MY THOUGHTS AND OPINIONS CHANGE FROM TIME TO TIME; THIS IS A NATURAL OFFSHOOT OF HAVING AN OPEN AND INQUISITIVE MIND.
SECURITY DRAGON @ NETSUITE

THE VIEWS AND OPINIONS EXPRESSED HERE ARE MY OWN ONLY AND IN NO WAY REPRESENT THE VIEWS, POSITIONS OR OPINIONS - EXPRESSED OR IMPLIED - OF MY EMPLOYER (PRESENT AND PAST) OR ANYONE ELSE. MY THOUGHTS AND OPINIONS CHANGE FROM TIME TO TIME; THIS IS A NATURAL OFFSHOOT OF HAVING AN OPEN AND INQUISITIVE MIND.
Ladies and gentlemen, I’d like to present to you - the Internet!
Ladies and gentlemen, I’d like to present to you - the Internet!
Ladies and gentlemen, I’d like to present to you - the Internet!
NO ONE SAID IT WAS SECURE
NO ONE SAID IT WAS SECURE
OOPS!
NO ONE SAID IT WAS SECURE
OOPS!
“OPEN SOURCE IS MORE SECURE.”

OOPS!
"OPEN SOURCE IS MORE SECURE."

INTERNET SOCIETY PRESIDENT
EVERYBODY’S JOB IS NOBODY’S JOB
“This is a story about four people named
“This is a story about four people named Everybody,
“This is a story about four people named
Everybody,
Somebody,
“This is a story about four people named

Everybody,
Somebody,
Anybody, and
“This is a story about four people named

Everybody,

Somebody,

Anybody, and

Nobody.
“This is a story about four people named

Everybody,
Somebody,
Anybody, and
Nobody.
There was an important job to be done and Everybody was asked to do it.
Everybody was sure
Somebody would do it.
Anybody could have done it, but
Nobody did it.
Somebody got angry about that,
because it was
Everybody's job.

EVERYBODY'S JOB IS NOBODY'S JOB
Everybody thought

EVERYBODY’S JOB IS NOBODY’S JOB
Anybody could do it but
Nobody realized that
Everybody wouldn't do it.
It ended up that Everybody blamed Somebody when Nobody did what Anybody could have done.”
It ended up that Everybody blamed Somebody when Nobody did what Anybody could have done.”
It ended up that Everybody blamed Somebody when Nobody did what Anybody could have done.”
It ended up that Everybody blamed Somebody when Nobody did what Anybody could have done.”

The bug revealed this week was buried inside 10 lines of code and would have been spotted in an audit, according to Laurie, who works on the security team at Google. "Unless you're doing an audit, you're not going to see it," he said.
“OPEN SOURCE PROJECTS PLAY A CRUCIAL ROLE IN THE DIGITAL AGE BUT ARE MAINTAINED BY A SMALL, STRAINED CADRE OF VOLUNTEERS.”
Performance

Stability

Security
Performance

Stability
"THERE ARE LOTS OF CRITICAL LIBRARIES MAINTAINED BY VOLUNTEERS THAT ARE NOT GIVEN ENOUGH ATTENTION"
“WHEN YOU CARRY POINTERS AROUND AND CANNOT TRACK WHETHER THEY ARE ALIVE AND HOW LONG THEY ARE, IT'S GOING TO HURT.”
“IT IS NOT LIKE JAVA GOT INSECURE ALL OF A SUDDEN. IT HAS BEEN INSECURE FOR YEARS.”
SOMETHING'S FUCKY
NO STRATEGY
NO SECURITY CONTACT OR REPORTING DEFINED
Inconsistent coding styles, usage, or complex code
“System administrators hate change when they have to bear the brunt of adverse effects of change.”
“System administrators hate change when they have to bear the brunt of adverse effects of change.”
WHAT DO WE DO NOW?
DEMOS
final String nnAddr = req.getParameter(JspHelper.NAMENODE_ADDRESS);
if (nnAddr == null){
  out.println(JspHelper.NAMENODE_ADDRESS + " url param is null");
  return;
}

DFSClient dfs = getDFSClient(ugi, nnAddr, conf);
String target = dir;
final HdfsFileStatus targetStatus = dfs.getFileInfo(target);
if (targetStatus == null) { // not exists
  out.println("<h3>File or directory : " + StringEscapeUtils.escapeHtml(target) + ">
  JspHelper.printGotoForm(out, namenodeInfoPort, tokenString, target,
  nnAddr);

public static void printGotoForm(JspWriter out,
    int namenodeInfoPort,
    String tokenString,
    String file,
    String nnAddress) throws IOException {
  out.println("<form action="browseDirectory.jsp" method="get" name="goto">");
  out.println("<title>");
  out.println("Goto :");
  out.println("<input type="text" width="50" id="dir" value="" + HtmlQuoting.quoteHtmlChars>
  out.println("<input type="submit" value="go"/>");
  out.println("<input name="namenodeInfoPort" type="hidden" value="" + namenodeInfoPort + ">");
  if (UserGroupInformation.isSecurityEnabled()) {
    out.println("<input name="" + DELEGATION_PARAMETER_NAME
    + " type="hidden" value="" + tokenString + ">");
  }
  out.println("<input name="" + NAMENODE_ADDRESS + " type="hidden" value=""");
  out.println("<input name="" + nnAddress + ">");
  out.println("</form>");
108 final String nnAddr = req.getParameter(JspHelper.NAMENODE_ADDRESS);
109 if (nnAddr == null){
110     out.print(JspHelper.NAMENODE_ADDRESS + " url param is null");
111     return;
112 }
113 
114 DFSClient dfs = getDFSClient(ugi, nnAddr, conf);
115 String target = dir;
116 final HdfsFileStatus targetStatus = dfs.getFileInfo(target);
117 if (targetStatus == null) { // not exists
118     out.print("<h3>File or directory : " + StringEscapeUtils.escapeHtml(target) + " does not exist";
119     JspHelper.printGotoForm(out, namenodeInfoPort, tokenString, target,
120         nnAddr);
452 public static void printGotoForm(JspWriter out,
453         int namenodeInfoPort,
454         String tokenString,
455         String file,
456         String nnAddress) throws IOException {
457     out.print("<form action="\browseDirectory.jsp" method="get" name="\goto">";
458     out.print("Goto ");
459     out.print("<input name="dir" type="text" width="50" id="dir" value="" + HtmlQuoting.quoteHtmlChars;
460     out.print("<input name="go" type="submit" value="go"/>");
461     out.print("<input name="namenodeInfoPort" type="hidden" 
462         + "value="" + namenodeInfoPort + "/>");
463     if (UserGroupInformation.isSecurityEnabled()) {
464         out.print("<input name="" + DELEGATION_PARAMETER_NAME
465             + "" type="hidden" value="" + tokenString + "/>");
466     }
467     out.print("<input name="" + NAMENODE_ADDRESS + "/" type="hidden" 
468         + "value="" + nnAddress + "/>");
469     out.print("</form>");
470 }
memset(ht->buffer, 0, 1024);
ht->table = (struct mypasswd **) rad_malloc (tablesize * sizeof(struct mypasswd *));
if (!ht->table) {
    /*
    * Unable allocate memory for hash table
    * Still work without it
    */
    ht->tablesize = 0;
    return ht;
}
memset(ht->table, 0, tablesize * sizeof(struct mypasswd *));
while (fgets(buffer, 1024, ht->fp)) {
    if(*buffer =='
' && (!ignorenewline || (*buffer != '+' && *buffer != '-'))) {
        if(!hashentry = mypasswd_malloc(buffer, nfields, &len))

static struct mypasswd * mypasswd_malloc(char const* buffer, int nfields, size_t* len) {
    struct mypasswd *t;
    /* reserve memory for (struct mypasswd) + listflag (nfields * sizeof (char*)) +
    ** fields (nfields * sizeof (char*)) + strlen (inst->format) + 1 */
    *len=sizeof (struct mypasswd) + nfields * sizeof (char*) + nfields * sizeof (char )
    t = (struct mypasswd *) rad_malloc(*len);
    /* Allocate memory, or exit.
    * This call ALWAYS succeeds!
    */
    void *rad_malloc(size_t size) {
        void *ptr = malloc(size);
        if (ptr == NULL) {
            ERROR("no memory");
            fr_exit(1);
        }
        return ptr;
Free Radius 3.0.3

```
/*
 * @todo: This code is SHIT. It's badly formatted. It's hard to understand. It re-implements tons of things which are already in the server core.
 */

memset(ht->buffer, 0, 1024);
ht->table = (struct mypasswd **) rad_malloc (tablesize * sizeof(struct mypasswd *));
if (!ht->table) {
    /*
    * Unable allocate memory for hash table
    * Still work without it
    */
    ht->tablesize = 0;
    return ht;
}
memset(ht->table, 0, tablesize * sizeof(struct mypasswd *));
while (fgets(buffer, 1024, ht->fp)) {
if (*buffer && *buffer!='/\n' && (!ignorenis || (*buffer != '+' && *buffer != '-'))) {
    if (!hashentry = mypasswd_malloc(buffer, nfields, &len)) {
        /* Allocate memory, or exit.
         * This call ALWAYS succeeds!
         */
        void *rad_malloc(size_t size)
        { void *ptr = malloc(size);
          if (ptr == NULL) {
            ERROR("no memory");
            fr_exit(I);
          }
          return ptr;
        }
```

```
typedef struct filter_cmd {
    struct filter_cmd *next;
    int command;
    BOOL seen;
    BOOL noerror;
    union argtypes args[1]; //8 bytes
} filter_cmd;

if (addr->reply->once_repeat < 0)
{
    *error_pointer = string_sprintf("Bad time value for \"once_repeat\" 
        "in mail or vacation command: %s",
        commands->args[mailarg_index_once_repeat]); //96 bytes
    return FF_ERROR;
}

```c
typedef struct filter_cmd {
    struct filter_cmd *next;
    int command;
    BOOL seen;
    BOOL noerror;
    union argtypes args[1]; //8 bytes
} filter_cmd;

if (addr->reply->once_repeat < 0)
{
    *error_pointer = string_sprintf("Bad time value for \"once_repeat\" ",
        "in mail or vacation command: %s",
        commands->args[mailarg_index_once_repeat]); //96 bytes
    return FF_ERROR;
}
```
I AM LEGION
(AND SO CAN YOU!)
STREET CRED

LULZ

FULL DISCLOSURE

WHAT IS YOUR INCENTIVE?
TUNING AND CORRELATION
1: APPLY: $\mathcal{G}$ to $\mathcal{P}$ to create subpopulations: $\{\mathcal{P}_{\hat{g}_1}, \ldots, \mathcal{P}_{\hat{g}_m}\}$
2: for all $\mathcal{P}_{\hat{g}_i}$ do
3: \begin{align*}
\mathcal{P}_{\hat{g}_i}, \hat{p} & \leftarrow \mathcal{P}_{\hat{g}_i}, s / \mathcal{P}_{\hat{g}_i}, n \\
\text{if } \max \mathcal{P}_{\hat{g}_i}, n < 51 \text{ then} \end{align*}
4: \begin{align*}
p_0 & \leftarrow \left(\frac{1}{m}\right) \sum_{i=1}^{m} \mathcal{P}_{\hat{g}_i}, \hat{p} \\
\text{else} \end{align*}
5: \begin{align*}
p_0 & \leftarrow 0.85 \\
\text{for all } \mathcal{P}_{\hat{g}_i} \text{ do} \end{align*}
6: \begin{align*}
\text{if } \mathcal{P}_{\hat{g}_i}, s = 0 \text{ and using NO-SUCCESS HEURISTIC then} \end{align*}
7: \begin{align*}
\text{Discard } \mathcal{P}_{\hat{g}_i} \end{align*}
8: \begin{align*}
\text{else} \end{align*}
9: \begin{align*}
\mathcal{P}_{\hat{g}_i}, z & \leftarrow (\mathcal{P}_{\hat{g}_i}, \hat{p} - p_0)/\sqrt{p_0(1 - p_0)/\mathcal{P}_{\hat{g}_i}, n} \\
\text{CREATE equivalence classes } \{E_{z_1}, \ldots, E_{z_k}\}: E_{z_1} & \leftarrow \{\mathcal{P}_{\hat{g}_j} \cap \mathcal{E}_{\mathcal{P}_{\hat{g}_j}, z = z_1}\} \\
\text{SORT } \{E_{z_1}, \ldots, E_{z_k}\} \text{ by } z_i \text{ in descending order. Designate that order as } \\
\{E_{z(i)}, \ldots, E_{z(k)}\} \end{align*}
10: \begin{align*}
\text{for all } i = 1, \ldots, k \text{ do} \end{align*}
11: \begin{align*}
\text{Inspect error reports in } E_{2(i)} \text{ using an auxiliary ranking scheme} \end{align*}
TUNING AND CORRELATION

1: APPLY $\mathcal{G}$ to $\mathcal{P}$ to create subpopulations: $\{\mathcal{P}_{\mathcal{G}_1}, \ldots, \mathcal{P}_{\mathcal{G}_m}\}$
2: for all $\mathcal{P}_{\mathcal{G}_i}$ do
3:   $\mathcal{P}_{\mathcal{G}_i}, \hat{p} \leftarrow \mathcal{P}_{\mathcal{G}_i}, s/\mathcal{P}_{\mathcal{G}_i}, n$
4: if $\max_{\mathcal{P}_{\mathcal{G}_i}, n < 5}$ then
5:   $p_0 \leftarrow \left(\frac{1}{n}\right) \sum_{i=1}^{m} \mathcal{P}_{\mathcal{G}_i}, \hat{p}$
6: else
7:   $p_0 \leftarrow 0.85$
8: for all $\mathcal{P}_{\mathcal{G}_i}$ do
9:   if $\mathcal{P}_{\mathcal{G}_i}, s = 0$ and using NO-SUCCESS HEURISTIC then
10:      Discard $\mathcal{P}_{\mathcal{G}_i}$
11: else
12:      $\mathcal{P}_{\mathcal{G}_i}, z \leftarrow (\mathcal{P}_{\mathcal{G}_i}, \hat{p} - p_0)/\sqrt{p_0(1 - p_0)/\mathcal{P}_{\mathcal{G}_i}, n}$
13: CREATE equivalence classes $\{E_{z_1}, \ldots, E_{z_k}\}$: $E_{z_i} \leftarrow \mathcal{P}_{\mathcal{G}_j} \cap \mathcal{E}[\mathcal{P}_{\mathcal{G}_j}, z = z_i]$
14: SORT $\{E_{z_1}, \ldots, E_{z_k}\}$ by $z_i$ in descending order. Designate that order as $\{E_{z_{(1)}}, \ldots, E_{z_{(k)}}\}$.
15: for all $i = 1, \ldots, k$ do
16:      Inspect error reports in $E_{z_{(i)}}$ using an auxiliary ranking scheme

1: $I \leftarrow H$
2: while $U \neq \emptyset$
3:   for each $r$ in $U$
4:      $r, prob \leftarrow \text{UPDATEPROBABILITY}(r, I, \mathcal{M})$
5:      $r, info \leftarrow \text{INFOGAIN}(r, U, I, \mathcal{M})$
6:      $r, best, value \leftarrow \text{FIND-MAX-BY-PROB-THEN-INFO}(U, \epsilon)$
7:      $r, best, value \leftarrow \text{USER-INSPECT-REPORT}(r, best)$
8:      $I \leftarrow I \cup \{r, best\}$
9:      $U \leftarrow U \setminus \{r, best\}$
- hacktheplanet.ninja/index.html
ONE MORE THING...

KEEP CALM AND HIDE THE INTERNET
2788: char name[PATH_MAX]; // 4,096 Bytes

2802: fscanf is 33,554,431 bytes