

# The Patriot Missile Failure

Berühmt-berüchtigte Softwarefehler Seminar SS 2003  
bei Dr. Bernard Beckert  
Universität Koblenz-Landau

Martin Eisemann  
200210279

13. November 2003

# Inhaltsverzeichnis

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>Einführung</b>	<b>3</b>
<b>3</b>	<b>Das Patriot System</b>	<b>3</b>
3.1	Aufbau . . . . .	3
3.2	Historisches . . . . .	4
3.3	Funktionsweise . . . . .	4
<b>4</b>	<b>Der Fehler</b>	<b>5</b>
4.1	Wann versagte das Patriot System . . . . .	5
4.2	Der wirkliche Fehler und Bugfixes . . . . .	6
4.3	Verbesserungsvorschläge . . . . .	7
4.4	Warum hat keiner die Verbesserungen anständig vorgenommen .	9
<b>5</b>	<b>Weiteres Versagen</b>	<b>9</b>
<b>6</b>	<b>Patriot Einsatz Eignung</b>	<b>11</b>
<b>7</b>	<b>Epilog</b>	<b>11</b>
<b>A</b>	<b>Fehlerursachen und Gegenmaßnahmen</b>	<b>12</b>

# 1 Abstract

Während des Sommersemesters 2003 wurden an der Universität Koblenz-Landau, im Rahmen des Seminars *Berühmt berüchtigte Softwarefehler* bei Herrn Dr. B. Beckert, verschiedene Softwarebugs der letzten Jahrzehnte vorgestellt, welche durch ihre Ausmaße, sei es nun in finanzieller oder sogar physischer Hinsicht, z.B. Verlust von Menschenleben, bekannt geworden sind. In dieser Arbeit soll das Versagen der Patriot Rakete im Golfkrieg 1991 behandelt werden.

# 2 Einführung

Am 25. Februar 1991, während des ersten Golfkrieges, versagte ein Patriot-Luftabwehrsystem bei Dhahran, Saudi-Arabien, in seiner Aufgabe eine irakische Scud-Rakete zu erfassen und abzufangen. Die Folge war, dass diese ungehindert in eine amerikanische Barracke einschlug und dabei 28 Soldaten tötete und ca. 100 verletzte.

Der Grund, wie sich später ergab, sollte ein Softwarefehler im Waffenkontrollcomputer sein. Ein Fehler, der je länger das System operierte, sich stets aufakkumulierte und so zu drastischen Fehlberechnungen führen konnte. Ob dieser allerdings allein schuld am Versagen des Systems war, wie er überhaupt entstehen konnte und was dagegen unternommen hätte werden können, soll im Weiteren näher betrachtet werden.

# 3 Das Patriot System

## 3.1 Aufbau

Das Patriot-Luftabwehrsystem ist ein mobiles Boden-Luft-Verteidigungssystem gegen Flugzeuge und Cruise Missiles, welches erstmalig in den 1960ern eingesetzt wurde. Damals sollte es der Verteidigung in Europa gegen sowjetische Angriffe dienen.

In neuerer Zeit, seit ca. 1990, wurde es zudem darauf getrimmt auch sogenannte 'short-range ballistic missiles' abzufangen. Welche sich mit einer vergleichsweise äusserst hohen Geschwindigkeit bewegen.

Das Patriot System selbst besteht aus einem Battalion, welches normalerweise aus sechs separaten Batterien besteht. Jede dieser Batterien besteht aus einer Radareinheit, welche für Feinderkennung und -verfolgung zuständig ist. Acht Missile Launcher dienen dem Abfangen. Eine Engagement Control Station dient als Schnittstelle für einen Benutzer, um etwa gefundene Ziele zu bestätigen.

Das zentrale Modul des Patriot Systems ist jedoch der Weapon Control Computer, welcher sämtliche Hauptaufgaben des Verfolgens und Abfangens durchführt, sowie noch einige andere Funktionen.

## 3.2 Historisches

Der erste Einsatz des Patriot Luftabwehr Systems fand Mitte der 1960er (1964) in Europa zur Verteidigung gegen sowjetische Angriffe statt. Die Ziele bewegten sich dabei mit Geschwindigkeiten von bis zu MACH2, also ca. 1500 mph.

Die Verteidigung gegen Short-Range Ballistic Missiles wurde erst später, Anfang der 90er hinzugefügt. Größter Unterschied zu vorherigen Zielen war die wesentlich gesteigerte Geschwindigkeit von bis zu MACH5, also ca. 3750 mph.

Hauptaugenmerk beim Entwurf des Systems waren vor allem Mobilität und kurze Einsatzzeiten an einem Ort, allerdings sollte damit wohl eher gemeint sein, dass das System schnell gestartet und beendet werden kann und nicht, dass es nach einer gewissen Zeit nicht mehr korrekt arbeitet. Sollte ersteres der Fall sein, könnte man sich natürlich sofort fragen wieso nicht ein Modul eingebaut wurde, welches das System regelmäßig neubootet?

## 3.3 Funktionsweise

Im Folgenden soll zunächst das standarmäßige Verhalten des Patriotsystems bei der Erkennung und sein Abfangmechanismus erläutert werden.

Ein Radar (bzw. mehrere) sendet regelmäßige Impulse aus, welche den Himmel nach Flugkörpern absキャンen. Trifft einer dieser Impulse auf ein Objekt, wird er reflektiert und vom Radar wieder aufgefangen. Dieser kann daraus Informationen wie etwa Breitengrad, Längengrad und Höhenwinkel (Azimuth), vom Radar aus gesehen, auslesen. Da diese Informationen mehrfach zu unterschiedlichen Zeitpunkten aufgenommen werden, lassen sich daraus Informationen über die Flughöhe des Objekts, Geschwindigkeit, Richtung und Position ausrechnen. Diese Berechnungen scheinen jedoch nur unzureichend genau zu sein, da nachdem ein Objekt, was einem feindlichen Angreifer entsprechen könnte, entdeckt wurde, erst der eigentliche Test durch den sogenannten Range-Gate Algorithmus ausgeführt wird. Dieser muss jedoch vorher explizit auf eine bestimmte Art von Ziel eingestellt sein, etwa Flugzeug, Cruise Missile oder Short-Range Ballistic Missile. Andere Objekte werden sofort in der ersten Überprüfung verworfen. Wurde ein in Frage kommendes Objekt entdeckt, das die entsprechenden Charakteristiken aufweist, so bestimmt der Range-Gate Algorithmus, wo der Radar als nächstes nach dem Objekt suchen soll. Befindet sich das Objekt auch dort im anvisierten Bereich, dem sogenannten Range-Gate, so wird eine Bestätigung ausgesendet. Diese kann entweder manuell von einem Soldaten abgesegnet werden oder automatisch, was zur Folge hat, dass entsprechende Raketen zum Abfangen losgeschickt werden, sofern das Objekt in Reichweite liegt.

Der Radarradius des Patriot Systems beträgt etwa 70 km. Eine Scud fliegt, wie bereits erwähnt mit ca. 3750 mph. Das bedeutet, für Entdecken, Erkennen und Abfangen einer Scudmissile bleiben gerademal etwa eine halbe Minute. Ziemlich wenig wenn man bedenkt, was für eine Verantwortung auf den Leuten liegt, die den Abfang bestätigen müssen, falls das System im manuellen Modus läuft.

## 4 Der Fehler

### 4.1 Wann versagte das Patriot System

Aus israelischen Tests war bereits lange vor dem schicksalhaften 25. Februar bekannt, dass das System Fehler aufwies, sollte es über längere Zeit in Betrieb sein. Die Israelis entdeckten dabei eine zwanzig prozentige Verschiebung des Range-Gates von seiner Idealposition, sobald das System für gerade mal acht Stunden lief. Bei einer fünfzig prozentigen Verschiebung des Range-Gates würde das Patriot System eine Scud nicht mehr als solche erkennen, was bereits nach ca. 20 Stunden der Fall wäre.

Um nun herauszufinden, wie diese Verschiebung zustande kommen konnte, muss man sich zunächst einmal die 'Range-Gate Prediction' etwas näher betrachten. Diese ist im Prinzip eine Funktion, welche unter anderem die beiden Eingabeparameter Geschwindigkeit und Zeitpunkt benötigt, um die neue Range-Gate Position zu berechnen. Die Geschwindigkeit wird dabei von einer Floatingpointzahl repräsentiert, der Zeitpunkt hingegen wird über die interne Uhr in Integern gezählt, nämlich jeweils in Zehntelsekunden. Für die Berechnung muss die Integerzahl in eine Floatingpointdarstellung umgerechnet werden, indem sie mit 0.1 multipliziert wird. Da für die Repräsentation jedoch nur 24 Bit Register zur Verfügung standen, kam es unweigerlich zu Rundungsfehlern, da die Binärdarstellung von 0.1 einer unendlichen Reihe entspricht, die jedoch maschinenbedingt nach 24 Stellen abgeschnitten wird.

Zur näheren Erläuterung:  $1/10$  entspricht  $1/2^4 + 1/2^5 + 1/2^8 + 1/2^9 + 1/2^{12} + 1/2^{13} + \dots$  oder  $0.0001100110011001100110011\dots$  in Binärdarstellung.

Der "offizielle" Fehler lag nun in der Umrechnung der als Integer gezählten Zehntelsekunden zu einer 24bit-Floating-Point Darstellung. Warum "offiziell" wird später nochmal aufgegriffen.

Zum besseren Verständnis soll hier ein ähnliches, vereinfachtes Beispiel aufgeführt werden auf einem five digit Register, dass die Drittel einer Sekunde zählt. Das heißt es stehen lediglich fünf Ziffern zur Darstellung einer Zahl zur Verfügung, inklusive ihrer Nachkommastellen.

- Bei Beginn

2	0.66667
1	<u>0.33333</u>
	0.33334

Fehler von  $-6.\bar{6} \cdot 10^{-6}$  Sekunden nach gerademal zwei Dritteln einer Sekunde.

- Etwas später

43	14.333
42	<u>14.000</u>
	0.333

Fehler von  $3.\bar{3} \cdot 10^{-4}$  Sekunden nach etwas mehr als 14 Sekunden.

- Viel Später

$$\begin{array}{r} 563 \quad 187,67 \\ 562 \quad \underline{187,33} \\ 0,34 \end{array}$$

Der Fehler ist bereits auf  $-6.\bar{6} \cdot 10^{-3}$  Sekunden angewachsen nach gerade mal knapp 188 Sekunden, bzw. ca. drei Minuten.

Damit sollte deutlich gemacht werden, dass sich die fehlerhafte Präzision der Range-Gate Berechnung direkt proportional zur Betriebsdauer des Patriotsystems verhält. Nämlich exakt  $2^{-20}$  pro Zehntelsekunde.

Die fehlerhafte Voraussage läßt den Radar die falsche Stelle überprüfen und sorgt dafür, dass feindliche Flugkörper möglicherweise nicht mehr als solche erkannt werden. Die Alpha-Batterie, welche die Dhahran Air Base verteidigen sollte, war bereits über hundert Stunden in Folge aktiv. Was einen Fehler von über 0.3433 Sekunden verursachte.

Die Berechnung lässt sich dabei folgendermaßen nachvollziehen: Binärdarstellung eines Zehntels auf einem 24-Digit-Register, bzw. als entsprechender Bruch:

$$0.00011001100110011001100_2 = \frac{209715}{2097152}$$

Berechnung des Fehlerwertes:

$$\left(\frac{1}{10} - \frac{209715}{2097152}\right)(100 * 60 * 60 * 10) = \frac{5625}{16384} \approx 0.3433sec$$

Eine Scud fliegt in dieser Zeit über einen halben Kilometer. Was zur Folge hat, dass sie sich damit bereits ausserhalb des Range-Gates befand.

Für den geneigten Leser ist hier ein kleines Beispielprogramm aufgelistet, welches den Fehler nachstellt. Das Ergebnis ist dabei allerdings maschinenabhängig und je nach verwendeter Programmiersprache nicht nachvollziehbar, mathematisch orientierte Sprachen wie Matlab oder Octave sind also nicht dafür geeignet.

Bsp.Code:

```
tenth = 1.0 / 10.0;
sec = 100 * 60 * 60 * 10;
t1 = sec * tenth;
t2 = sec/10.0;
output = t2 - t1;
```

## 4.2 Der wirkliche Fehler und Bugfixes

Der oben genannte Fehler wurde am 11. Februar 1991 durch die Israelis bekannt, also bereits zwei Wochen vor dem Angriff auf Dhahran. Die Reaktion der Obrigkeit auf die Erkenntnisse der Israelis war doch eher verwunderlich. Man sagte einfach, dass der normale Patriotbenutzer das System nicht länger als acht Stunden am Stück laufen lassen würde. Trotzdem brachte man am 16.

Februar ein Softwareupdate heraus, welches den Fehler beseitigen sollte. Durch die Kriegssituation schien es jedoch absehbar zu sein, dass dieses die verschiedenen Armeestationen im Krisengebiet nicht mehr rechtzeitig erreichen würde. Aus diesem Grunde schickte das Patriot Project Office am 21. Februar eine Nachricht an alle Patriotnutzer, in der geschrieben stand, dass zu lange Betriebszeiten einen Fehler in der Zielerfassung des Systems herbeiführen würde. Ein Softwareupdate sei allerdings unterwegs. Allerdings wurde mit keinem Wort darauf eingegangen, wie lange denn nun die Zeitspanne eigentlich sei, was man also unter einer langen Betriebsdauer verstehen dürfte. Wie sich zeigte schien die Auffassung darüber doch ein wenig zu variieren. Ausserdem waren die israelischen Daten auch nie wirklich ausgewertet worden, das heißt man wusste eigentlich nicht, wie lange das System nun wirklich laufen durfte, bevor es seine Funktion nicht mehr ausführen konnte. Interessanterweise traf das Softwareupdate genau einen Tag nach der Katastrophe in Dhahran ein.

Aber eigentlich erklärt all dies in keinster Weise, wie der Fehler zustande kommen konnte. Denn genaugenommen wird ja lediglich die Zeit zwischen zwei Radarimpulsen benötigt. Da es sich dabei lediglich um Sekunden handeln darf fällt der Fehler dabei eigentlich unmerklich klein aus, sofern die Differenz noch auf Integern durchgeführt wird. Berechnungen haben ergeben, dass die Abweichung bei 0,0001 % liegen muss. Zu klein um wirklich gefährlich zu sein. Aber was hat dann den Fehler so groß werden lassen? Schließlich war das doch die offizielle Erklärung.

Zunächst einmal, die Software, welche zur Berechnung benutzt wurde, war vor bereits dreissig Jahren in Assembler geschrieben worden, als man noch nicht soviel Ahnung davon hatte, wie man Berechnungen genauer ausführen konnte. Damit das Patriotsystem nun fähig sein sollte Short-Range Ballistic Missiles abzufangen, wurden mehrere Softwareupdates herausgegeben, unter ziemlichem Zeitdruck, weil die Operation Desert Storm ja bereits anlief. Zumindest eine davon enthielt eine Subroutine, welche die Umrechnung der internen Zeitrechnung in die Floating-Point-Repräsentation genauer vornehmen sollte. Indem ein Paar von 24-Bit Registern genutzt werden sollte zur Repräsentation der Zeit, anstatt nur einem. Diese Subroutine wurde an ungefähr einem halben Dutzend Stellen im Code eingebaut, dummerweise nicht an allen. Die aktuelle Zeit wurde dadurch mit der alten Methode berechnet, also relativ ungenau, während das Tracken der Missiles, das Zielen und die Berechnung, wann genau die Abwehrrakete abgefeuert werden soll, mit der neueren, akkurateren Berechnung durchgeführt wurde. Das System nutzte aber die alte Darstellung der Zeit um zu testen, ob die SCUD zum gegebenen Zeitpunkt in der Range Gate lag, was unwiderrufflich zum entsprechenden Fehler führen musste.

### 4.3 Verbesserungsvorschläge

Es stellt sich einem nun die Frage, wessen Fehler war es eigentlich? Mensch oder Maschine? Die Maschine, weil eine relativ ungenaue Berechnung durchgeführt wurde? Aber ein menschliches Versagen hat den Fehler ja erst herbeigeführt, durch die nicht ausreichend geprüften Bugfixes. Oder war es letzten Endes doch

die schlechte Informationsweitergabe, sprich keine genaue Angabe wie lange das System fehlerfrei laufen würde? Reichte es bekanntzugeben, dass das System nicht dauerhaft funktionierte? Warum keine genauen Zeitangaben für die Soldaten, damit sie wissen wann ein Neustart unumgänglich ist? Das Hauptaugenmerk beim Entwurf des Systems waren ja, wie bereits erwähnt, Mobilität und kurze Einsatzzeiten an einem Ort, aber was für ein Stil ist es denn, bereits bei der Planung numerische Fehler einzugestehen, und warum wurde dann kein Modul miteingeplant, das das System regelmäßig neu bootet? Auch wenn es dadurch alle acht Stunden etwa für ca. ein bis zwei Minuten nicht einsatzfähig wäre, so wäre das nur das kleinere Übel gegenüber der Unfähigkeit seine Funktion zu erfüllen, sollte es länger am Stück aktiv sein.

Es stellt sich einem die Frage, was sonst noch hätte unternommen werden können um den Fehler abzuschwächen? Den Code-Fix schneller schicken? War angeblich durch die Kriegssituation ja nicht möglich. Automatisches rebooten des Systems alle acht Stunden? Ziemlich schlecht, wenn in diesem Zeitrahmen gerade eine Rakete angefliegen kommt. Allerdings, mit wachsendem Wissen über die Scuds, hätte man errechnen können, ab wann das System einfach nutzlos gegenüber diesen Raketen ist, so dass ein Neubooten einfach unumgänglich wäre. Vielleicht Warnlichter, damit die Soldaten manuell entscheiden können ob ein Neustart gerade möglich wäre? Etwas schlecht, wenn der Feind praktisch sehen kann, wann er angreifen darf. Er bräuchte ja nur einen Spion mit Fernglas auf dem nächstgelegenen Berg hocken haben, der die Lampe beobachtet.

Bessere Spezifikationen, damit solche Fehler wie mit der Subroutine nicht auftreten können, wären natürlich eine Möglichkeit gewesen. Oder zumindest die Dokumentation verbessern, damit man weiss, wo der geänderte Programmteil noch überall Einfluss drauf hat. Das Programm hätte auch modularer gehalten werden können, das heißt, dass einzelne Komponenten besser ersetzt und ausgetauscht werden können. War mit Assembler allerdings noch nicht wirklich möglich. 1986 jedoch wurde eine grössere Änderung am Code vorgenommen, um das System dazu zu befähigen auch Cruise Missiles und nicht nur Flugzeuge abzufangen. Diese Chance hätte man vielleicht nutzen können, um den Code komplett neu in einer High Level Programminglanguage schreiben zu können, da man in Friedenszeiten auch nicht unter solchem Zeitdruck steht. Das wäre im Endeffekt nicht nur sicherer gewesen, sondern letzten Endes vielleicht sogar günstiger, da bei schlechtdesignter Software häufig immense Kosten in der Wartung entstehen, aber das kommt hier ja darauf an, was Raytheon, der Hersteller, seinen Kunden versprochen hat. Ausserdem wäre in Notzeiten der Code einfacher und schneller anzupassen.

Bessere und mehr Verifikationsverfahren. Dagegen spricht allerdings Zeitdruck und wohl auch Budget, denn die Umschulung auf Short-Range Ballistic Missiles erfolgte ja erst kurz vor dem Golfkrieg, so dass ausreichende Testverfahren nicht möglich waren. Hätte der Fehler denn überhaupt gefunden werden können, wenn man bei Raytheon mehr Tests durchgeführt hätte? Wahrscheinlich ja, aber jeder dieser Tests bringt immense Kosten auf. In den Anfangsphasen des Projektes wurde ein zwanzigstündiger Kampf nachgestellt, bei dem wirkliche Missiles und Flugzeuge benutzt wurden. Die Kosten kann man sich vorstellen. Für jeden ein-

zelen Patch einen erneuten Kampf darzustellen dürfte jede Firma in den Ruin treiben.

#### 4.4 Warum hat keiner die Verbesserungen anständig vorgenommen

Der Weapon Control Computer, das Herz des Patriot Systems basiert auf einem Design aus den Siebzigern. Das heißt nicht unbedingt, dass es schlecht ist, aber beschränkte Kapazitäten des damaligen Systems könnten dafür gesorgt haben, dass sich hier nicht nur einige numerische Schwachstellen eingeschlichen haben könnten. Das heißt fehlende Präzision war nur eine der Schwierigkeiten mit denen die Entwickler zu kämpfen hatten. Zudem war das gesamte System zunächst in Assembler geschrieben worden, was das ganze auch nicht gerade wartbar macht.

Genaugenommen war das Patriotsystem zum Zeitpunkt seines Einsatzes, während der Operation Desert Storm, bei weitem noch nicht ausgereift. Im Gegenteil, es war das erste Mal, dass es überhaupt für den Einsatz gegen Scud Missiles getestet wurde. Das Vorgehen war dabei eher experimentell als wirklich fundiert geplant. Man hatte eigentlich noch keine wirklichen Informationen über das Flugverhalten der Scuds. Alles was man später eingebaut hatte, erfuhr man über Erfahrungen im Feld und Geheimdienste, wobei sich hier die Frage stellt, wie zuverlässig Informationen aus diesen Quellen sind. Auch die Felderfahrungen erwiesen sich als schwerer verwertbar als es hätte sein müssen. Denn dummerweise gab es im Patriot System keine eingebauten Datenrekorder und externe Rekorder waren den amerikanischen Kommandeuren zu gefährlich, da sie einen unerwarteten Systemabsturz im Patriot System herbeiführen könnten. Die Folge war, dass nur israelische Einheiten sich scheinbar getraut hatten Datenrekorder zu verwenden. Die so gewonnenen Informationen stellten sie dann der U.S. Army zur Verfügung.

Die Testphase für die Patriot musste so kurz ausfallen, weil es als ein schlimmeres Übel angesehen wurde den Soldaten gar keinen Schutz zu liefern als ein System, dass nicht tadellos funktioniert. Von August 1990 bis Februar 1991 arbeitete man deswegen verbissen an den Änderungen des Patriot Systems, welche es für die Operation Desert Storm einsatzfähig machen sollten. So entstanden in dieser Zeit insgesamt sechs verschiedene Patches. Die Folge war, dass sich eigentlich niemand mehr wirklich mit dem kompletten Code auskannte. Dementsprechend war es auch nicht verwunderlich, dass sich die entscheidende Subroutine für die Zeitberechnung nicht ohne Probleme einbringen liess, bzw. nicht sorgfältig genug bedacht wurde, an welche Stelle sie einzufügen war.

## 5 Weiteres Versagen

Das Fiasko in Dhahran während des ersten Golfkrieges war nicht das einzige Mal, dass das Patriot System die Aufmerksamkeit der Öffentlichkeit auf sich zog. Auch wenn die nächste Meldung zunächst etwas auf sich warten ließ. Die

Menschen schienen sich nach den Unternehmungen zur Fehlerbehebung Anfang der neunziger zu sicher zu sein, dass jetzt alles tadellos funktionieren würde. Denn obwohl das System immer noch offiziell für Kurzeinsätze entworfen war, wurden einige Raketen, so etwa im persischen Golf und Südkorea, mehrere Jahre in höchster Alarmbereitschaft gehalten. Teilweise bis 10 Jahre am Stück. Scheinbar schien niemand zu begreifen, dass numerische Fehler zwar minimiert aber selten völlig entfernt werden können. So schien sich irgendwo im System noch ein kleiner Bug zu befinden, der dann allerdings erst nach einer wesentlich längeren Zeitspanne aktiv, bzw. bemerkbar wurde. Dieser sorgte dafür, dass die Raketen Probleme hatten eine Radio Frequenz zu empfangen, die ihnen Anweisungen während des Fluges zukommen ließ, so berichtet von BBC News am 24. März 2000.

Allerdings schien auch hier nicht nur das blinde Vertrauen schuld zu sein, sondern auch einfach fehlgeleitete oder verlorengegangene Information. Denn der Waffenhersteller Raytheon Co. hatte offiziell nur eine tadellose Betriebsdauer von sechs Monaten zugesichert. An einigen Standorten wurde dies auch einigermassen bedacht, allerdings wurden die Raketen dort nur einmal im Jahr ausgetauscht und nicht alle sechs Monate. Dieses seltsame Verhalten lässt sich eigentlich nur auf zwei Arten erklären. Entweder hat sich Raytheon Co. so schlecht ausgedrückt, dass man eher von einer längeren Betriebsdauer ausgehen konnte, wie auch die Militärobirgkeit damals ihren Soldaten gegenüber, oder die Information war mal wieder nicht an alle Verantwortlichen weitergeleitet worden, vielleicht auch beides. Oder man wollte sogar Schwächen verheimlichen, aus Sicherheitsgründen, was die US Army sogar offiziell zugab.

Für erneutes Aufsehen sorgte das Patriot System gerade noch vor wenigen Monaten. So sorgte eine Fehlfunktion in der Software dafür, dass ein Patriot-Raketensystem einen Tornado-Kampffjet der British Royal Airforce abgeschossen hat (so berichtet in TecChannel vom 28.6.2003). Offizielle Reaktionen, besagten jedoch, dass kein Softwarefehler sondern menschliches Versagen die Schuld trug, laut Verteidigungsindustrieexperten. Es säße immer ein Mensch noch am Kontrollzentrum, der die letztendliche Entscheidung trifft, es gäbe keinen Automatikmodus, der die Anordnung zum Abfeuern der Raketen geben könnte. Der Hersteller Raytheon allerdings wirbt auf seiner Webseite genau damit für das Patriotsystem.

Am Montag dem 23.6.2003 soll zudem ein amerikanischer Pilot das Radarsystem einer Patriot-Raketenbatterie zerstört haben, um dieses davon abzuhalten seinen bereits als feindlich eingestuften F-16 Kampffjet abzufangen. Das lässt nicht nur Rückschlüsse auf ein wirklich nicht verlässliches System schließen, sondern es lässt sich zudem scheinbar auch noch recht simpel zerstören.

Eigentlich dürfte es solche Zwischenfälle gar nicht geben, da in jedem Kampffjet ein sogenanntes Friend or Foe-System eingebaut ist. Dieses sendet regelmäßig ein Signal aus, welches dem Patriotsystem mitteilt, dass es sich hierbei nicht um einen Feind handelt. Damit der Feind dieses Signal natürlich nicht erfährt muss es allerdings während des Kampfeinsatzes ausgeschaltet werden. Leider vergaßen wohl die Piloten des häufigeren es während ihrer Rückkehr dann erneut zu aktivieren.

## 6 Patriot Einsatz Eignung

Aufgrund der häufig negativen Meldungen über das Patriot System haben sich einige Leute Gedanken darüber gemacht, ob das Patriotsystem überhaupt geeignet sei Scuds abzufangen. Auffällig dabei wurde, dass es keinen einzigen wirklich Beweis gibt, dass das Patriotsystem überhaupt in der Lage ist Scuds abzufangen, denn einiges spricht dagegen. Es existieren keine wirklichen Aufzeichnungen in denen ein erfolgreicher Abfang dokumentiert ist. Nur wenige Erfolgsmeldungen von Raytheon selbst natürlich. Angeblich werden gerade einmal zwischen null bis vier von insgesamt 45 Short Range Ballistic Missiles abgefangen. Theorien dafür gibt es genügend. So ist die Patriot Rakete zum Beispiel langsamer als eine Scud. Bei Versuchen galt es schon als Erfolg, wenn die Patriot den Pfad der Scud gekreuzt hat.

Theodore Postol vom MIT und Reuven Pedatzur aus Tel Aviv bewiesen durch unabhängige Untersuchungen, dass das Patriot System eine Erfolgsrate von unter zehn Prozent bzw. sogar null Prozent haben muss. Zum einen, weil nicht der Sprengkopf, sondern der Schwerpunkt der Scud anvisiert wird und zum anderen, weil die Scuds so schlecht verarbeitet sind, dass sie eine nahezu unvorhersehbare Flugroute haben, was ein Abfangen nahezu unmöglich macht.

Der Erfolg der Patriot Rakete war also eher psychologischer Art, ähnlich der scheinbar starken Bedrohung durch die Scud Raketen, welche auch niedriger war als befürchtet. Die Nutzer hatten einfach ein Gefühl von Sicherheit und Feinde wurden abgeschreckt. Wenn jedoch eine Scud ihr Ziel gefunden hat, waren die Auswirkung verheerend.

Das israelische Verteidigungsministerium war sogar so unzufrieden mit der Erfolgsquote des Patriotsystems, dass sie schon Anstrengungen unternommen hatten, selbst Angriffe gegen den Irak vorzunehmen, während des ersten Golfkrieges, was nur dadurch verhindert wurde, dass ein Waffenstillstand zustande kam.

Komischerweise verwenden sie es immer noch, in Verbindung mit Arrow Missiles um ihre Nuklearreaktoren und Sammelplätze ihrer Nuklearwaffen zu verteidigen.

## 7 Epilog

Trotzdem gaben die Amerikaner die Hoffnung auf die Patriot nicht auf und investierten seit der Operation Desert Storm über drei Milliarden US Dollar in das System. Vielleicht aus Überzeugung, dass es doch noch funktionieren könnte, vielleicht aber auch einfach nur aus Ermangelung einer Alternative.

## A Fehlerursachen und Gegenmaßnahmen

Datum	Fehlerursache	Gegenmaßnahmen
11.2.1991	Unpräzise Umrechnung von Integer- zu 24bit-Floating-Point Darstellung (offizieller Fehler)	Softwareupdate am 16.Feb. und Hinweis zur Benutzung am 21.Feb.
Aug. 1990 - Feb. 1991	Insgesamt sechs Software-patches wurden ausgeliefert, von denen eines eine Subroutine enthielt, die nicht an alle erforderlichen Stellen im Programmcode eingefügt wurde. (inoffizieller Fehler)	s.o.
im Kriegsverlauf	Fehlende Erfahrungen im Feld, fehlende Datenrekorder und zu wenig Informationen durch die Geheimdienste sorgten für schlechte Performanz des Systems.	Felderfahrung sammeln.
24.3.2000	Numerischer Fehler, der sich über Jahre hinweg aufakkumulierte und Probleme beim Empfangen der Steuerinformationen sorgte, sowie fehlgeleitete oder verlorengegangene Informationen über die maximale Betriebsdauer.	Austausch der entsprechenden Raketen.
28.6.2003	Fehlfunktion in der Software oder aber menschliches Versagen führten zum Abschluß eines Tornado Kampffjets der Royal British Airforce	Offziell wurde nichts unternommen, da menschliches Versagen die Ursache sein sollte.

## Literatur

- [Vanpul94] **Mark C. van Pul**(1994), A General Introduction to Software Reliability, CWI Quarterly Vol. 7(3)
- [Wong03] **Ken Wong**(2003), CMPUT 401: Lecture 17 Testing, Department of Computing Science University of Alberta
- [Demoura] **Demoura**, What is Rounding Error?, Universidade do Estado do Rio de Janeiro  
<http://magnum.ime.uerj.br/demoura/FAPERJ-SBM/arredondamento/roundtalk.ps>
- [Weisstein03] **Eric Weisstein**(1999-2003), Eric Weissteins's world of mathematics, Roundoff Error  
<http://mathworld.wolfram.com/RoundoffError.html>
- [Skeel92] **robert Skeel** (July 1992), Roundoff Error and the Patriot Missile, SIAM News, Vol. 25, Number 4, page 11  
<http://www.siam.org/siamnews/general/patriot.html>
- [Morgan02] **Tom Morgan an Jason Roberts** (2002), An Analysis of the Patriot Missile System  
<http://seeri.etsu.edu/SECCodeCases/ethicsC/PatriotMissile.htm>
- [Wik03] **Wikipedia - The Free Encyclopedia**(2003), MIM-104 Patriot  
[http://www.wikipedia.org/wiki/MIM-104\\_Patriot](http://www.wikipedia.org/wiki/MIM-104_Patriot)
- [Gao92] **GAO-Bericht B-247094**(1992), Patriot Missile Defense - Software problem led to system failure at Dhahran, Saudi Arabia
- [Can01] **George Candea, Armando Fox**(2001), Recursive Restartability: Turning the Reboot Sledgehammer into a Scalpel, 8th Workshop on Hot Topics in Operating Systems  
[http://roc.cs.berkeley.edu/recursive\\_restartability.pdf](http://roc.cs.berkeley.edu/recursive_restartability.pdf)
- [Muel01] **Norbert Th. Müller**(2001/2002), Vorlesung Rechner-Arithmetik im SS 2001/2002, Abteilung Informatik, Universität Trier  
<http://www.informatik.uni-trier.de/mueller/Lehre/arith-2002-06-18.pdf>
- [Tec03] **TecChannel uba**(2003), Software-Bug in Patriot-Systemen?  
<http://www.tecchannel.de/news/20030328/thema20030328-10329.html>
- [info03] **Paul Roberts**(2003), Software bug may cause Patriot Missile errors, Info World News, March 27, 2003  
[http://www.infoworld.com/article/03/03/27/HNpatriot\\_1.html](http://www.infoworld.com/article/03/03/27/HNpatriot_1.html)